Open Training Recipes for Reasoning in Language Models

Hanna Hajishirzi

PAUL G. ALLEN SCHOOL
OF COMPUTER SCIENCE & ENGINEERING

Ai2

AI is here today due to open scientific practices and fully open models

Are we done with scientific LM research and innovation?

Ai2

Research Still Needed

- Science of LMs
- Extend LMs Beyond Text
- Use LMs in Real World
- Improve LMs
- LMs for Science
- LM Agents
- Build Next generation of LMs
- LMs for Health
- Planning
- Test-time Inference
- Mitigate LMs Risk and Biases
- Efficient Models

Ai2

To facilitate innovation and accelerate the science of LMs

"AI institutes relying on proprietary models is like astronomy research about the solar system based on pictures printed in newspapers."

We need language models that are fully open.

Transparent

Reproducible

Accessible

Open Ecosystem to
Accelerate Innovation in Language Models

OLMo

Tulu

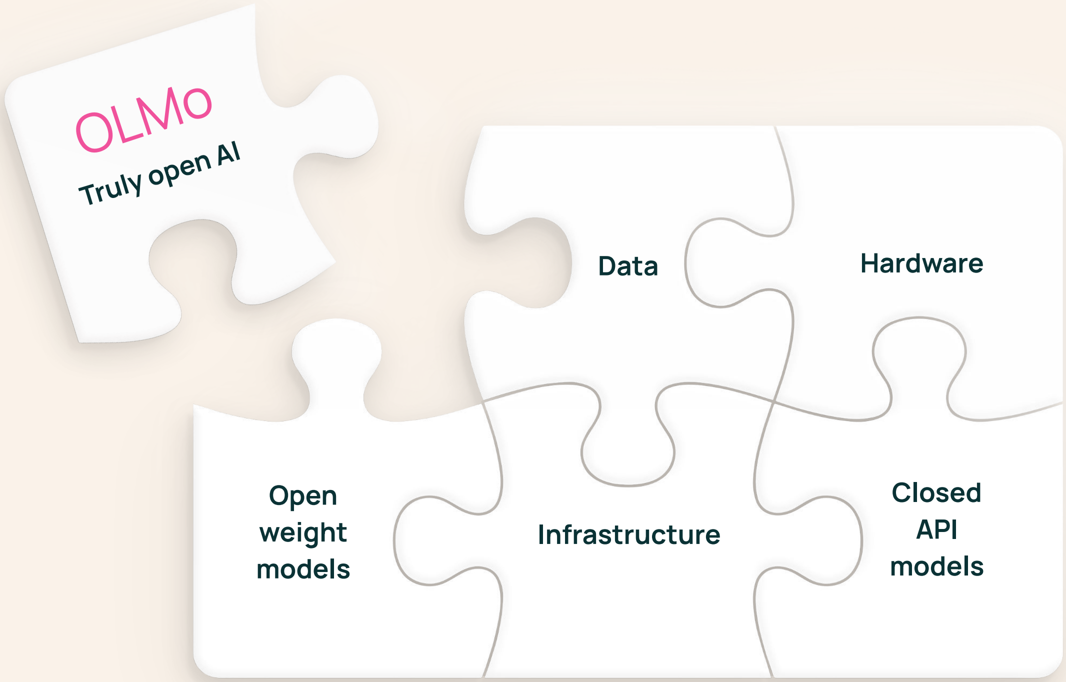
Fully open ecosystem

Develop, study, and advance LMs

Open, documented, and reproducible

Empower AI community

Public AI literacy



Ai2

Pre training

Post Training

Test-time
Inference

Many slides from:
Yizhong Wang, Nathan Lambert, Hamish Ivison, Faeze Brahman,
Niklas Muennighoff

Pre training

Post Training

Test time Scaling

OLMo

OLMo 2

OLMoE

Dolma

Tulu

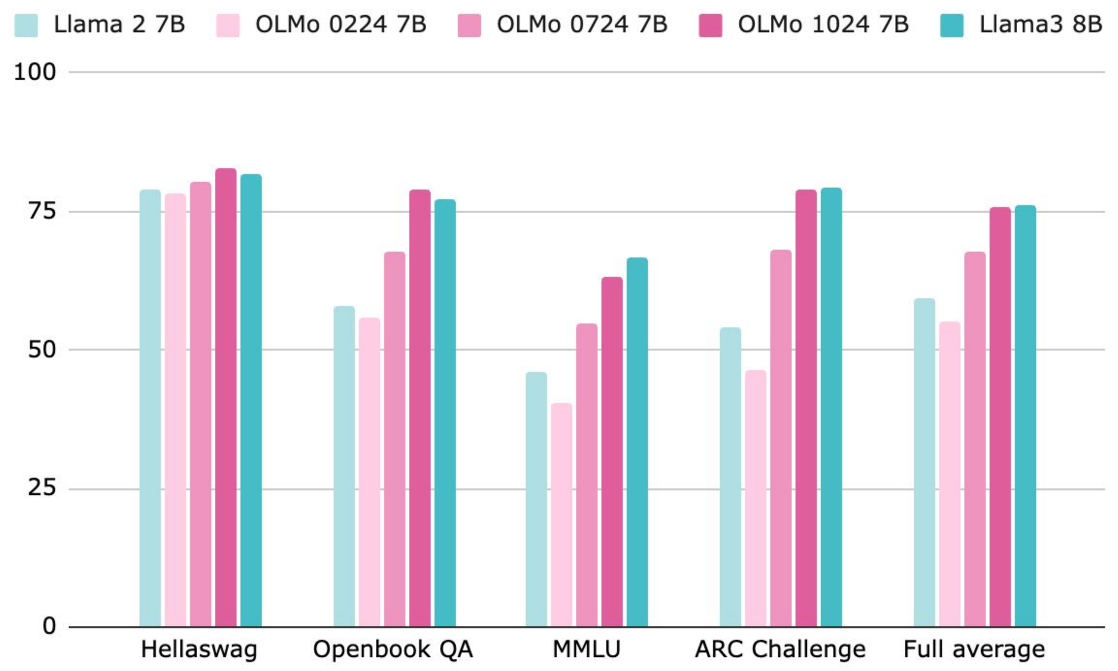
OLMo-Instruct

OpenInstruct Toolkit

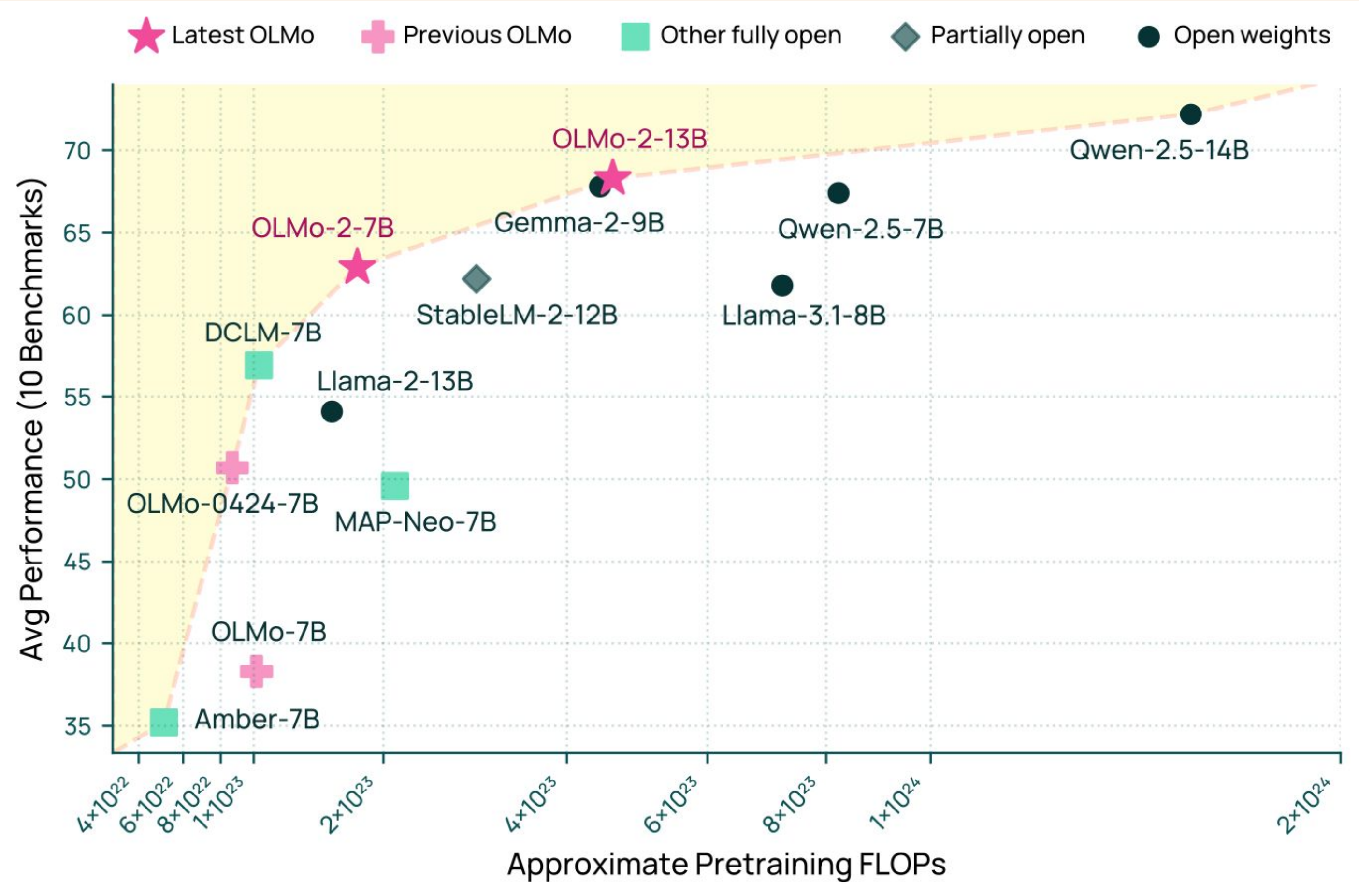
Safety Data & Toolkit

S1

OLMo 2



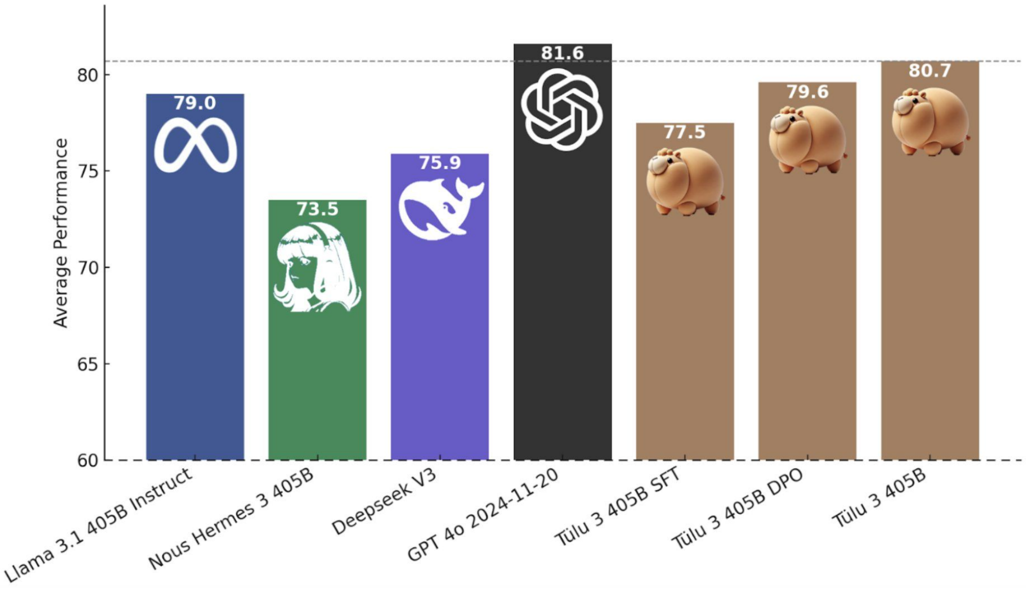
OLMo2 on par or better than Llama3, Qwen2.5



Latest OLMo
Previous OLMo
Other fully open
Partially open
Open weights
Avg Performance (10 Benchmarks)
70
65
60
55
50
45
40
35
OLMo-2-13B
Qwen-2.5-14B
Gemma-2-9B
Qwen-2.5-7B
OLMo-2-7B
StableLM-2-12B
Llama-3.1-8B
DCLM-7B
Llama-2-13B
OLMo-0424-7B
MAP-Neo-7B
OLMo-7B
Amber-7B
4×10^22
6×10^22
8×10^22
1×10^23
2×10^23
4×10^23
6×10^23
8×10^23
1×10^24
2×10^24
Approximate Pretraining FLOPs

Ai2

Tulu rivals DeepSeek and GPT4-o



Pre training

Post Training

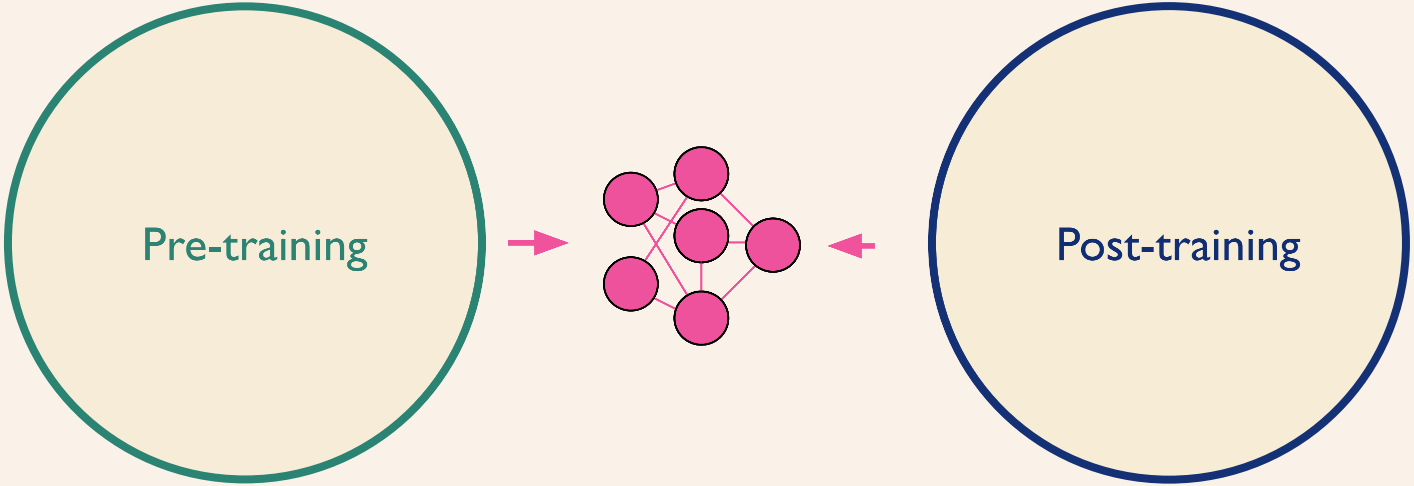
Test-time Inference

Tulu

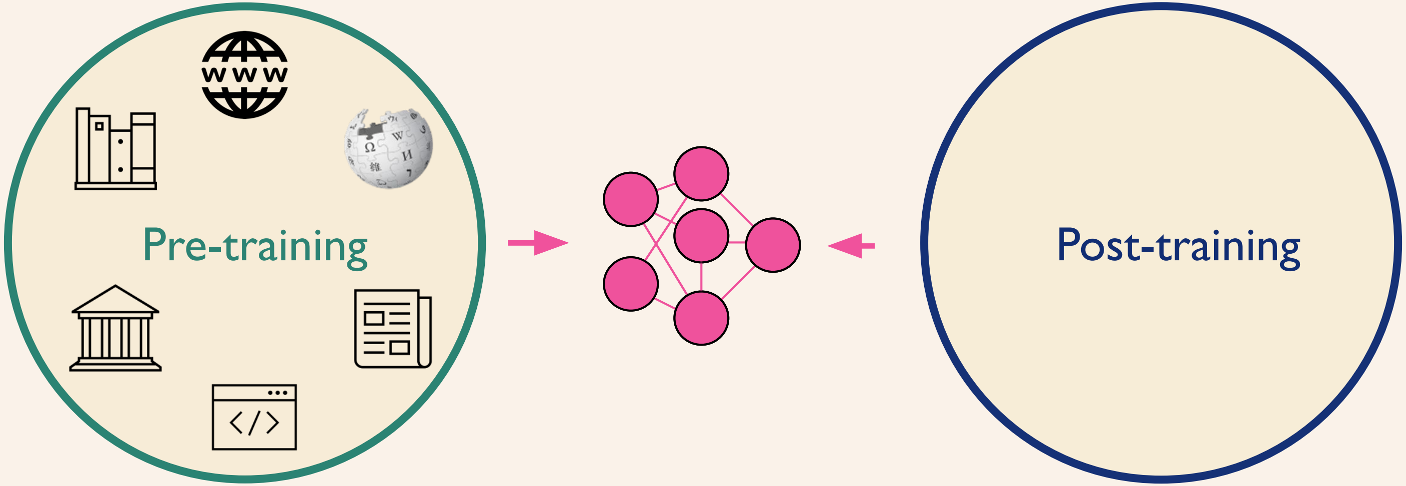
Many slides from:

Yizhong Wang, Nathan Lambert, Hamish Ivison, Faeze Brahman

Building a modern LLM

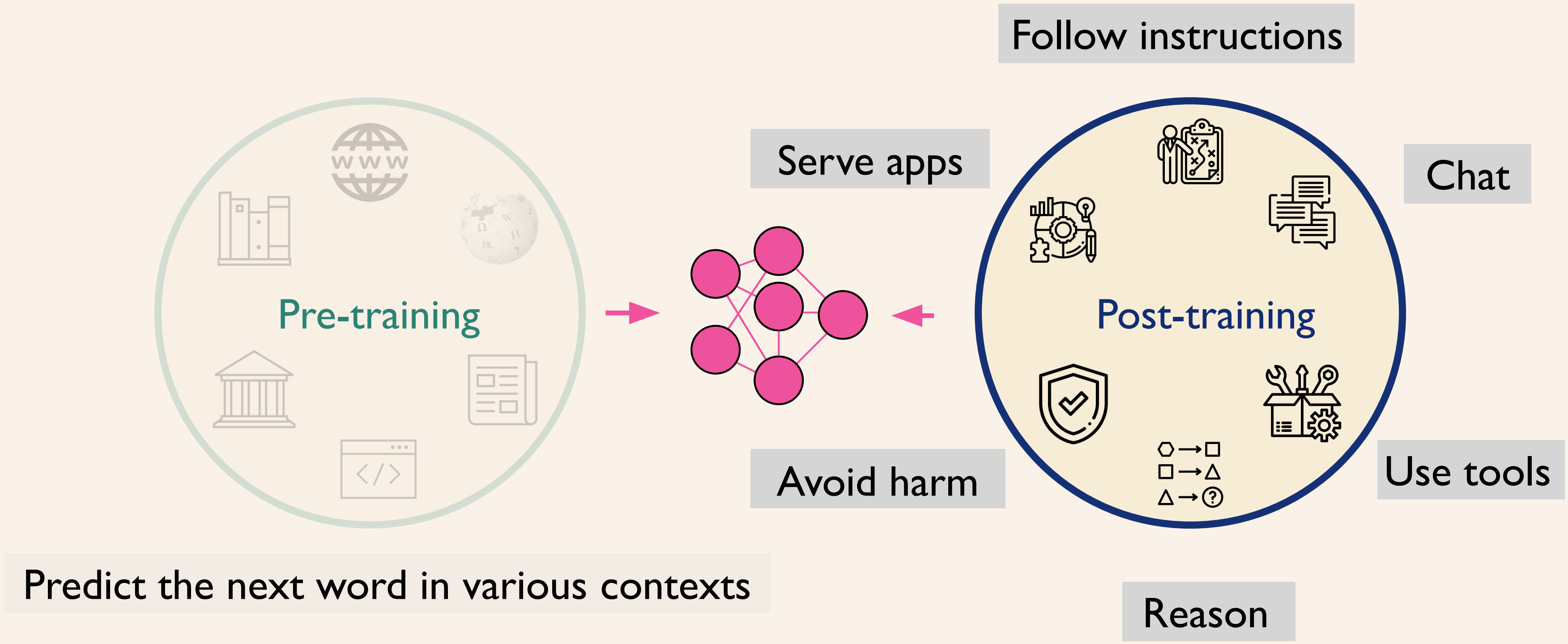


Building a modern LLM

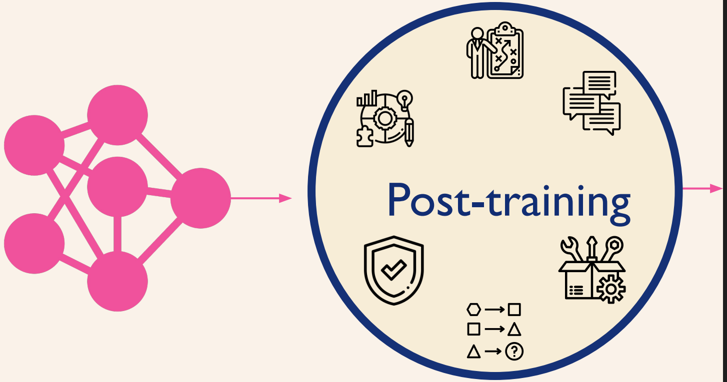


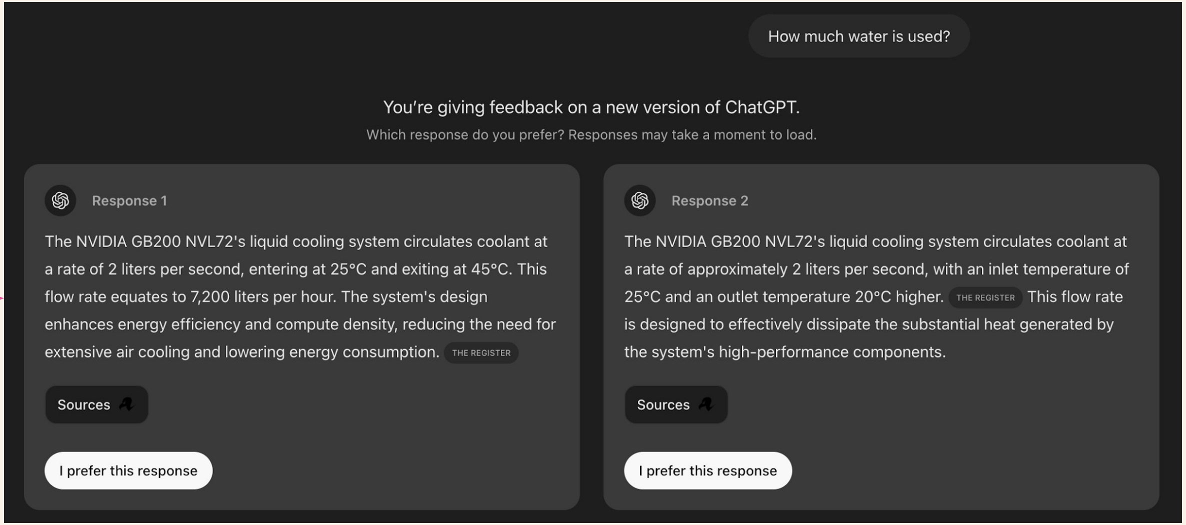
Predict the next word in various contexts

Building a modern LLM



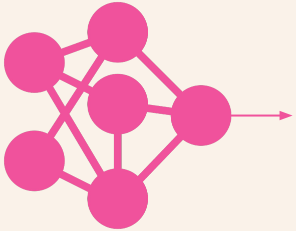
Post Training for Alignment with Human Preferences

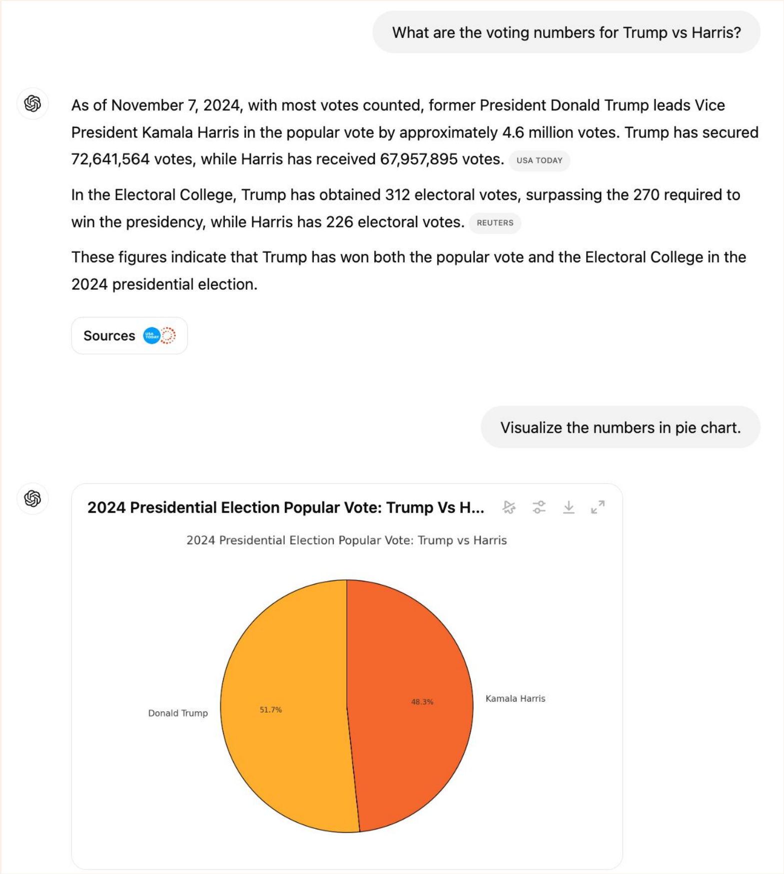




[Photo by @FanaHOVA on X]

Post Training for Tool Use /Agents



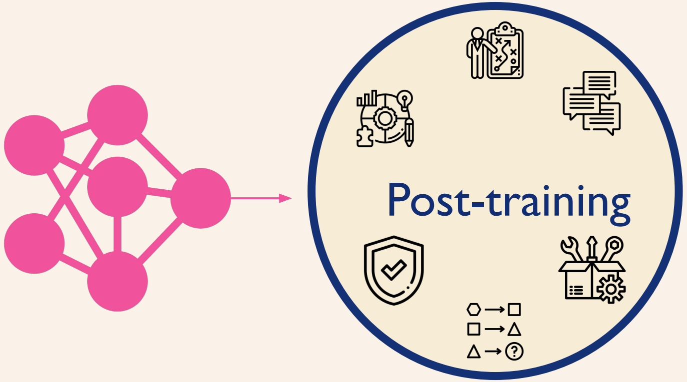


Searching

Code Execution

Ai2

Post Training for Reasoning



Find the smallest positive integer that can be expressed as the sum of two squares in two different ways.

To solve this problem, we need to find the smallest positive integer (n) that can be expressed as the sum of two squares in two different ways. This means we need (n) such that there exist distinct pairs ((a, b)) and ((c, d)) where:

[n = a^2 + b^2 = c^2 + d^2]

We will use Python to iterate through possible values of (n) and check if (n) can be expressed as the sum of two squares in two distinct ways.

Let's write a Python function to solve this problem step-by-step.

```
def find_smallest_sum_of_two_squares():
    def is_sum_of_two_squares(n):
        ways = set()
        for a in range(int(n ** 0.5) + 1):
            b_squared = n - a * a
            if b_squared >= 0:
                b = int(b_squared ** 0.5)
                if b * b == b_squared:
                    ways.add((a, b))
                    ways.add((b, a))
        return ways

    n = 1
    while True:
        ways = is_sum_of_two_squares(n)
        if len(ways) >= 2:
            return n, ways
        n += 1

smallest_n, ways = find_smallest_sum_of_two_squares()
print((smallest_n, ways))
```

```
(50, {(5, 5), (1, 7), (7, 1)})
```

The smallest positive integer that can be expressed as the sum of two squares in two different ways is (50).

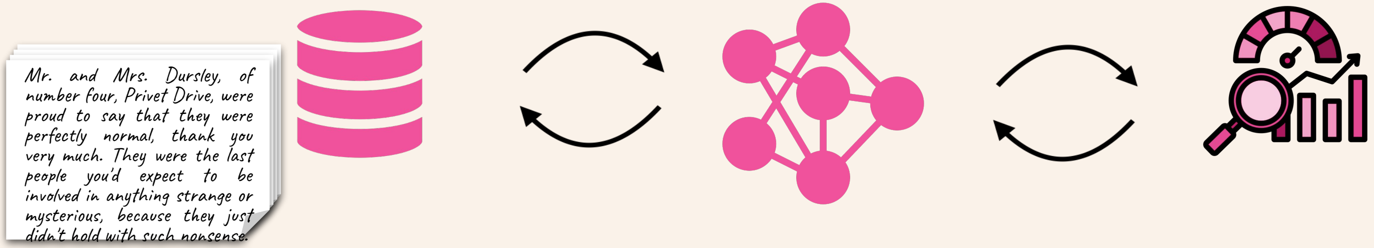
The ways to express 50 as the sum of two squares are: [50 = 1^2 + 7^2] [50 = 5^2 + 5^2]

Thus, the answer is (\boxed{50}).

Ai2

Building a modern LLM

Data | Models | Algorithms



Data — Training — Evaluation

Building a modern LLM

Data

- comes from different sources
- in different forms
- targets for different capabilities

How to use the right data in the right way?

Data

- comes from different sources
- in different forms
- targets for different capabilities

Tulu

Tülu

Open, reproducible, & state-of-the-art post-training recipe

[Wang*, Ivison* et al., 2023]

[Ivison*, Wang* et al., 2023]

[Ivison, Wang et al., 2024]

[Lambert, …, Wang, Dasigi, Hajishirzi, 2024]

Tulu: Open Instruction Tuning Recipe

How Far Can Camels Go? Exploring the State of Instruction Tuning on Open Resources

Yizhong Wang*♣♠ Hamish Ivison*♣ Pradeep Dasigi♣
Tushar Khot♣ Khyathi Raghavi Chandu♣ David Wadden♣ Ke
Noah A. Smith♣♠ Iz Beltagy♣ Hannaneh Hajish
♣Allen Institute for AI ♠University of Washington
{yizhongw,hamishi}@allenai.org

Best recipe for instruction data
Jun 2023

Camels in a Changing Climate: Enhancing LM Adaptation with TÜLU 2

Hamish Ivison*♠ Yizhong Wang*♣♠ Valentina Pyatkin♣♠
Matthew Peters♣ Pradeep Dasigi♣ Joel Jang ♣♠ Davi
Noah A. Smith♣♠ Iz Beltagy♣ Hannaneh Hajis
♣Allen Institute for AI ♠University of Washingto
{yizhongw,hamishiv}@cs.washington.edu

Best open model with preference data
Nov 2023

Unpacking DPO and PPO: Disentangling Best Practices for Learning from Preference Feedback

Hamish Ivison♣♠ Yizhong Wang♣♠ Jiacheng Liu♣♠
Zeqiu Wu♠ Valentina Pyatkin♣♠ Nathan Lambert♣
Noah A. Smith♣♠ Yejin Choi♣♠ Hannaneh Hajishirzi♣♠
♣Allen Institute for AI ♠University of Washington
hamishiv@cs.washington.edu

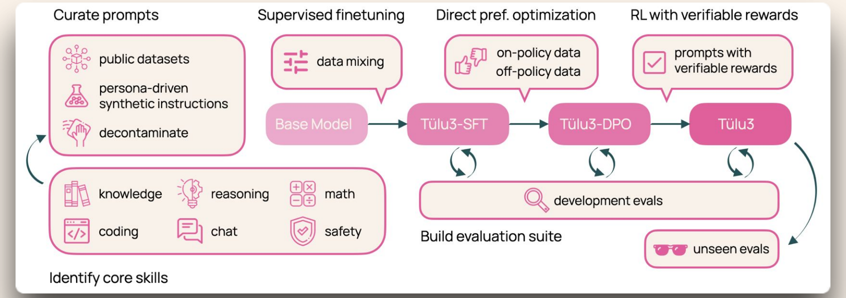
Systematic study of DPO vs PPO
June 2024

Open models & data

Tülu $1 \rightarrow 2 \rightarrow 2.5 \rightarrow 3$

Tülu 1
[Wang et al., NeurIPS 2023]

Open post-training recipe



Tülu 3 [Lambert et al., Arxiv 2024]

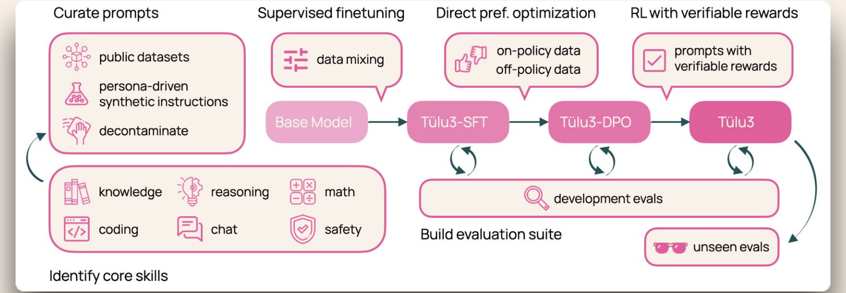
Ai2

Open models & data

Tülu 1→2→2.5→3

Tülu 1 [Wang et al., NeurIPS 2023]

Open post-training recipe



Tülu 3 [Lambert et al., Arxiv 2024]

Fully-open LM

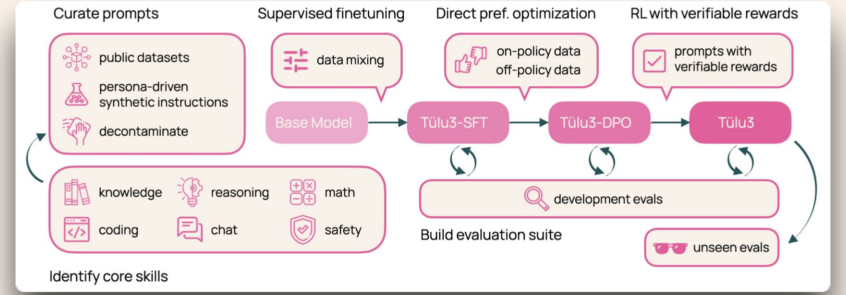
OLMo [Groeneveld et al., ACL 2024]

Open models & data

Tülu 1→2→2.5→3

Tülu 1 [Wang et al., NeurIPS 2023]

Open post-training recipe

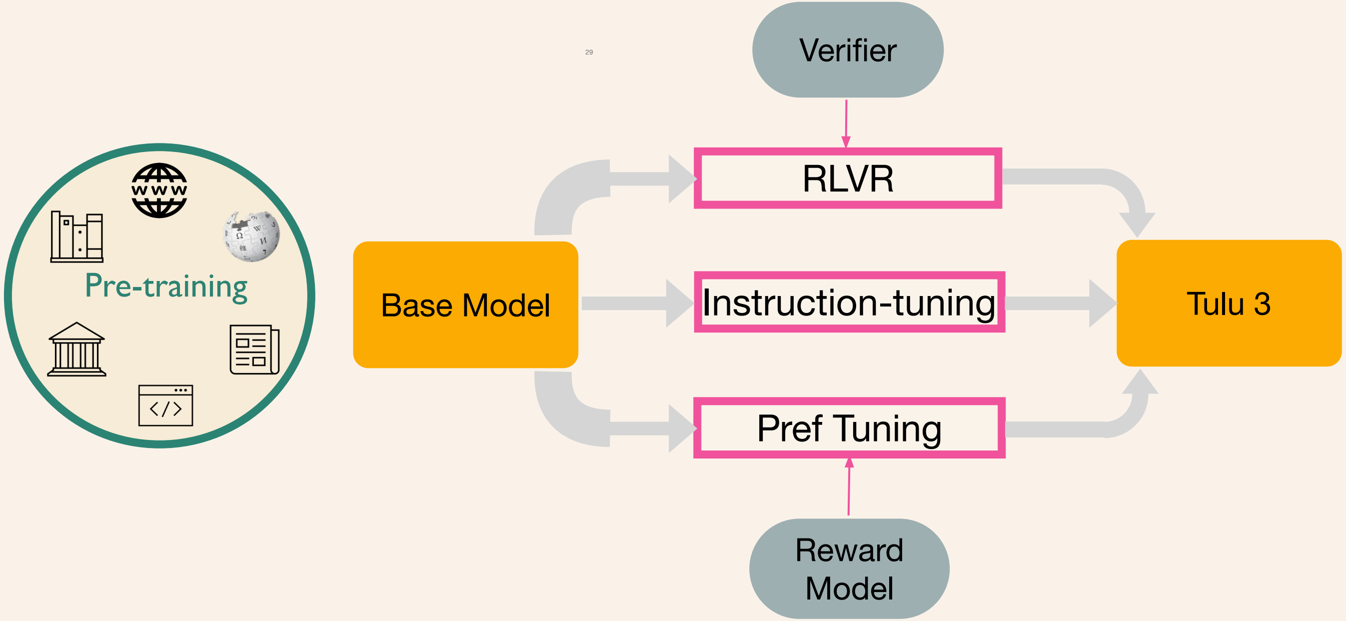


Tülu 3 [Lambert et al., Arxiv 2024]

Fully-open LM

OLMo [Groeneveld et al., ACL 2024]

Tülu 3 Training Recipe



29

Getting Ingredients to Start With

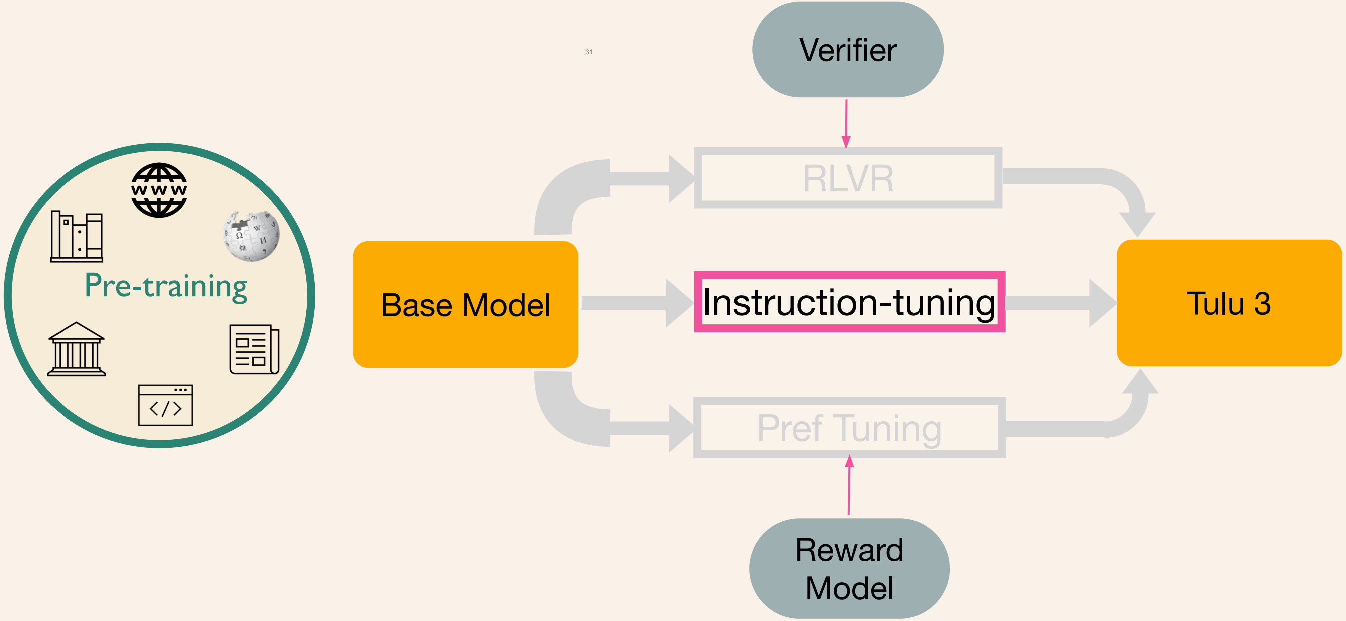
Successful adaptation starts with:

1. Meaningful **evaluations** for targeted skills
2. **Prompts** of representative queries for said skills
3. Check for Licenses
4. Decontamination

Category	Prompt Dataset	Count	# Prompts used in SFT	# Prompts used in DPO
General	**TÜLU 3 Hardcoded**[↑]	24	240	–
	OpenAssistant[1,2,↓]	88,838	7,132	7,132
	No Robots	9,500	9,500	9,500
	WildChat (GPT-4 subset)[↓]	241,307	100,000	100,000
	UltraFeedback[α,2]	41,635	–	41,635
Knowledge Recall	FLAN v2[1,2,↓]	89,982	89,982	12,141
	SciRIFF[↓]	35,357	10,000	17,590
	TableGPT[↓]	13,222	5,000	6,049
Math Reasoning	**TÜLU 3 Persona MATH**	149,960	149,960	–
	TÜLU 3 Persona GSM	49,980	49,980	–
	TÜLU 3 Persona Algebra	20,000	20,000	–
	OpenMathInstruct 2[↓]	21,972,791	50,000	26,356
	NuminaMath-TIR[α]	64,312	64,312	8,677
Coding	**TÜLU 3 Persona Python**	34,999	34,999	–
	Evol CodeAlpaca[α]	107,276	107,276	14,200
Safety & Non-Compliance	**TÜLU 3 CoCoNot**	10,983	10,983	10,983
	TÜLU 3 WildJailbreak[α,↓]	50,000	50,000	26,356
	TÜLU 3 WildGuardMix[α,↓]	50,000	50,000	26,356
Multilingual	Aya[↓]	202,285	100,000	32,210
Precise IF	**TÜLU 3 Persona IF**	29,980	29,980	19,890
	TÜLU 3 IF-augmented	65,530	–	65,530
Total		23,327,961	939,344	425,145

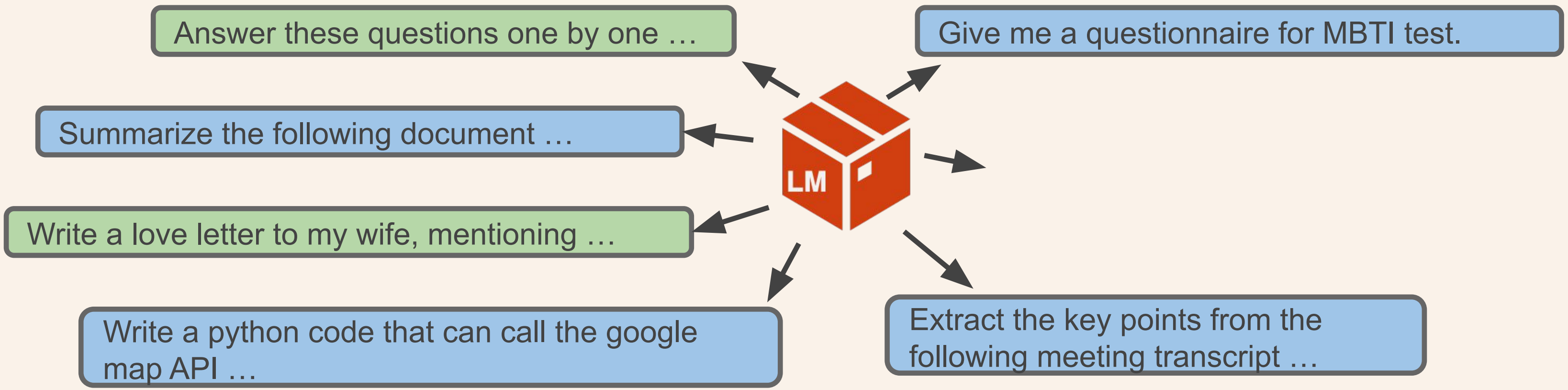
Ai2

Tülu 3 Supervised Finetuning (a.k.a Instruction Tuning)



Supervised Finetuning

- SFT (or Instruction tuning): Finetuning pretrained LMs with prompts and completions



Data Curation

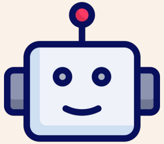
How to
source data?

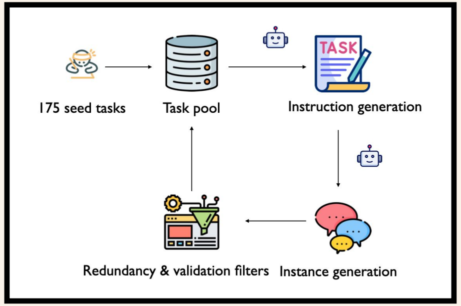
- Costly
- Time-consuming
- High variance

Data Curation

- Costly
- Time-consuming
- High variance

How to
source data?



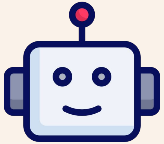


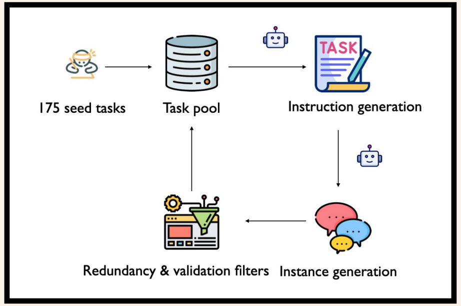
Self-Instruct [Wang et al., ACL 2023]

Synthetic data

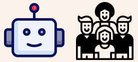
How to source data?

- Costly
- Time-consuming
- High variance





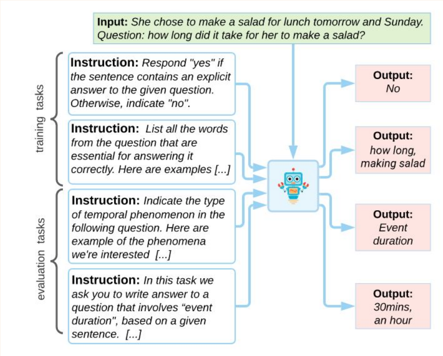
Self-Instruct [Wang et al., ACL 2023]

Combining  ?

Hybrid Preferences [Miranda*, Wang* et al., 2023]

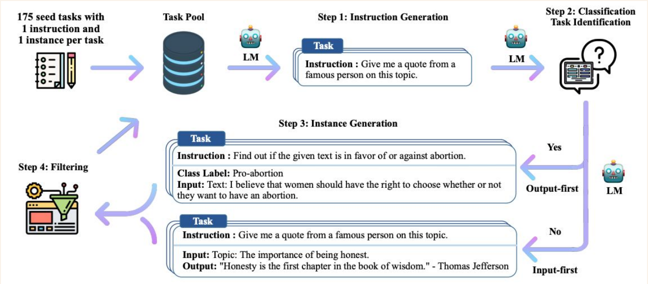
Ai2

Data Curation

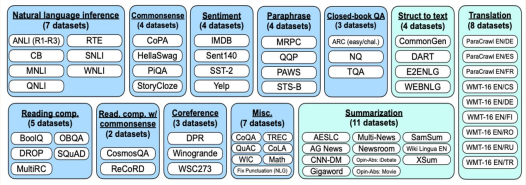


NaturalInstructions,
[Mishra et al 2022]

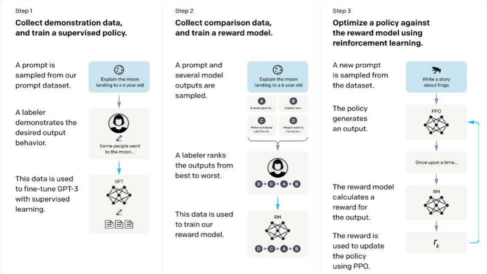
Super-NaturalInstructions,
[Wang et al. 2022]



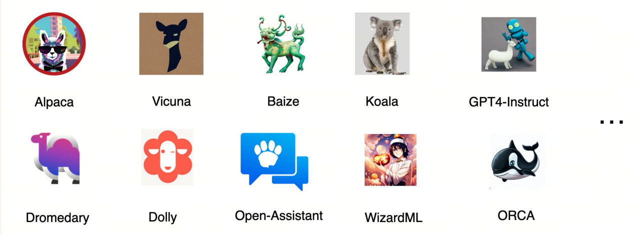
Self-Instruct,
[Wang et al. 2023]



FLAN_v1,
[Wei et al 2022]



InstructGPT,
[Wei et al 2022]



Lots of instruction datasets …

Supervised Finetuning: The role of data

Two repeated and parallelizable tracks:

1. **Data curation**: Curate data given targeted capabilities

2. **Data mixing**: Mix data across capabilities

   a. Substantial effort in filtering data while maintaining performance.
   b. Start fully with mixing before curation.

Tülu 1: instruction tuning data mixing

SuperNI
CoT
Flan V2
Dolly
Open Assistant 1
Self-instruct
Unnatural Instructions
Alpaca
Code-Alpaca
GPT4-Alpaca
Baize
ShareGPT

→ created by human

→ synthesized with GPT-3/4

Tülu 1: instruction tuning data mixing

	Chat (vibe)
SuperNI	4.2
CoT	6.0
Flan V2	3.2
Dolly	13.7
Open Assistant 1	58.1
Self-instruct	5.0
Unnatural Instructions	8.4
Alpaca	21.9
Code-Alpaca	15.8
GPT4-Alpaca	63.1
Baize	21.9
ShareGPT	70.5

Ai2

Tülu 1: instruction tuning data mixing

	Chat (vibe)	Knowledge	Reasoning	Coding	Multilinguality	Safety
SuperNI	4.2	49.7	4.3	12.9	50.2	22.7
CoT	6.0	44.2	41.0	23.7	47.8	56.1
Flan V2	3.2	50.6	30.4	16.8	47.2	38.6
Dolly	13.7	45.6	23.2	31.0	46.5	21.1
Open Assistant 1	58.1	43.3	27.3	31.9	33.4	94.8
Self-instruct	5.0	30.4	20.9	12.5	41.3	10.7
Unnatural Instructions	8.4	46.4	20.9	23.9	40.9	44.3
Alpaca	21.9	45.0	23.1	29.9	31.1	41.9
Code-Alpaca	15.8	42.5	24.6	34.2	38.9	8.0
GPT4-Alpaca	63.1	46.9	27.7	36.6	23.5	98.8
Baize	21.9	43.7	24.4	28.7	33.6	58.8
ShareGPT	70.5	49.3	33.7	34.1	30.5	97.5

Tülu 1: instruction tuning data mixing

	Chat	Knowledge	Reasoning	Coding	Multiling	Safety	Average
SuperNI	4.2	49.7	4.3	12.9	50.2	22.7	21.2
CoT	6.0	44.2	41.0	23.7	47.8	56.1	37.1
Flan V2	3.2	50.6	30.4	16.8	47.2	38.6	31.0
Dolly	13.7	45.6	23.2	31.0	47.2	21.1	29.2
Open Assistant 1	58.1	43.3	27.3	31.9	33.4	94.8	45.2
Self-instruct	5.0	30.4	20.9	12.5	41.3	10.7	20.2
Unnatural Instructions	8.4	46.4	20.9	23.9	40.9	44.3	29.4
Alpaca	21.9	45.0	23.1	29.9	31.1	41.9	30.8
Code-Alpaca	15.8	42.5	24.6	34.2	38.9	8.0	26.9
GPT4-Alpaca	63.1	46.9	27.7	36.6	23.5	98.8	46.3
Baize	21.9	43.7	24.4	28.7	33.6	58.8	33.6
ShareGPT	70.5	49.3	33.7	34.1	30.5	97.5	49.9
Human + Synthetic	56.5	49.3	41.9	35.9	45.6	99.9	53.0

What type of data is useful for reasoning?

- Response to reasoning queries are more than simple final answer

Question: A store has a "Buy 2 Get 1 Free" deal on shirts. Each shirt costs $25. If Sarah wants to buy 7 shirts, how much will she spend?

Chain-of-Thought Output:

Let me break this down:
1. With "Buy 2 Get 1 Free", every 3 shirts will cost the price of 2
2. 7 shirts can be grouped as: 2 complete sets (6 shirts) + 1 remaining shirt
3. For the complete sets: * 6 shirts = 2 × (2 paid + 1 free) * Cost = 2 × (2 × $25) = $100
4. For the remaining 1 shirt: * Cost = 1 × $25 = $25
5. Total cost = $100 + $25 = $125

Therefore, Sarah will spend $125

Requires step-by-step thought process (aka CoT)

Why Chain-of-Thought data for reasoning?

Chain of Thought data

👍🏼 helps models handle complex, multi-step problems easier

👍🏼 reveals the model's reasoning process

👍🏼 makes it easier to spot errors in logic thus more trustworthy

👍🏼 resembles human thought process

But ...

👎🏼 Manual annotation challenges:

- time and cost intensive
- often requires expert annotations
- Difficult to scale

Why Chain-of-Thought data for reasoning?

CoT ...

👍 helps models handle complex, multi-step problems easier

👍 reveals the model's reasoning process

👍 makes it easier to spot errors in logic thus more trustworthy

👍 resembles human thought process

But ...

👎 Manual annotation challenges:

- time and cost intensive
- often requires expert annotations
- Difficult to scale

Expensive
Time Consuming
Not diverse enough

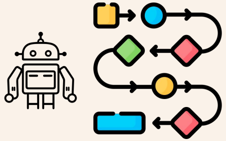
Our Approach: Hybrid Data Creation

Data mixing & selection from existing resources

Our approach: Hybrid Data Creation

Data mixing & selection from existing resources

+



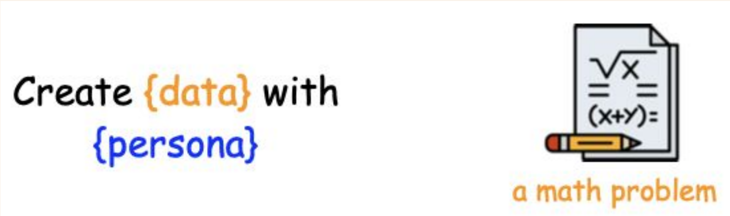
Persona-driven Data Synthesis

- Enable targeting specific skills (e.g., math, code, precise instruction following)
- Ensure high diversity
- Enable Scaling

Scaling Synthetic Data Creation with 1,000,000,000 Personas

Tao Ge,* Xin Chan, Xiaoyang Wang, Dian Yu, Haitao Mi, Dong Yu

Persona-driven Data generation for Scalability and Improved Diversity

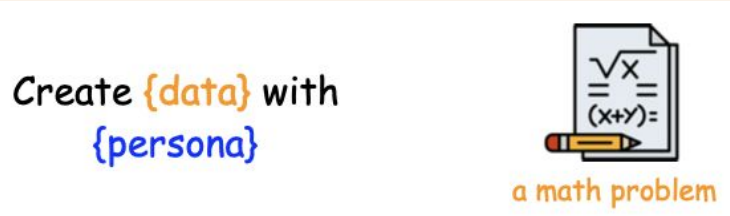


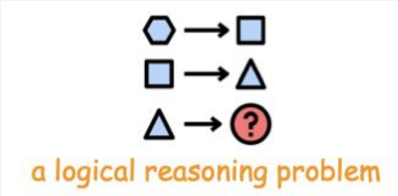
Dr. Smith, a chemist, is studying a reaction where compound X decomposes into products Y and Z. The reaction follows first-order kinetics with a rate constant k of 0.5 min^{-1}.

If the initial concentration of compound X is 1.0 M, how long will it take for the concentration of X to decrease to 0.25 M?

Photo from Ge et al. 2024

Persona-driven Data generation for Scalability and Improved Diversity





Dr. Smith, a chemist, is studying a reaction where compound X decomposes into products Y and Z. The reaction follows first-order kinetics with a rate constant k of 0.5 min^{-1}.

If the initial concentration of compound X is 1.0 M, how long will it take for the concentration of X to decrease to 0.25 M?

You are analyzing the spatial arrangement of molecules in a reaction chamber. There are three types: A, B, and C. Molecule A is always adjacent to B, but never to C. Molecule B can be adjacent to both A and C.

If molecule C is surrounded by other molecules, which ones must be present around it?

Photo from Ge et al. 2024

Persona-driven Data generation for Scalability and Improved Diversity

Create {data} with

~250K Personas

a math problem

~150k hard math problems

~50k grade school math problems

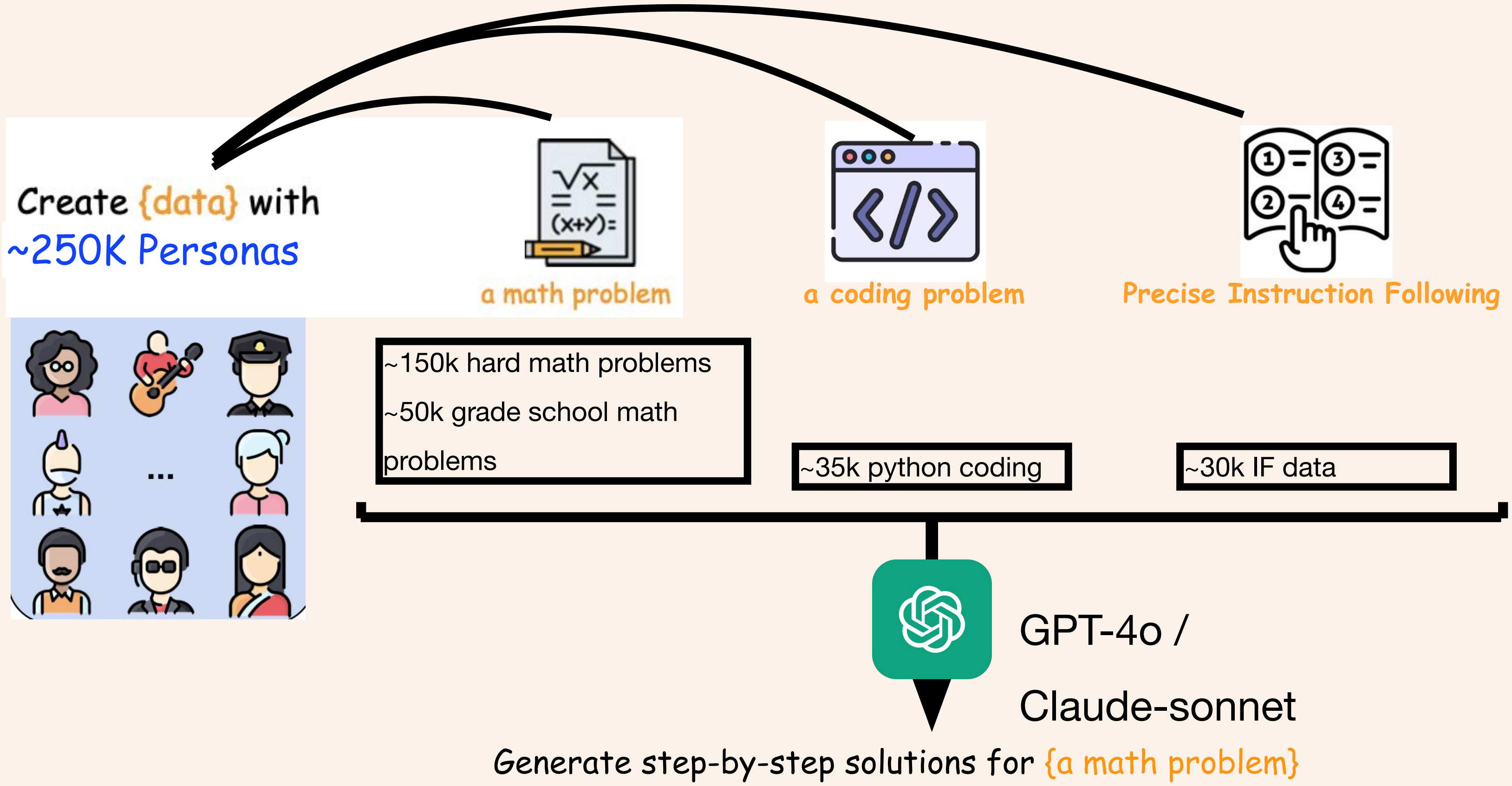
a coding problem

~35k python coding

Precise Instruction Following

~30k IF data

Persona-driven Data generation for Scalability and Improved Diversity

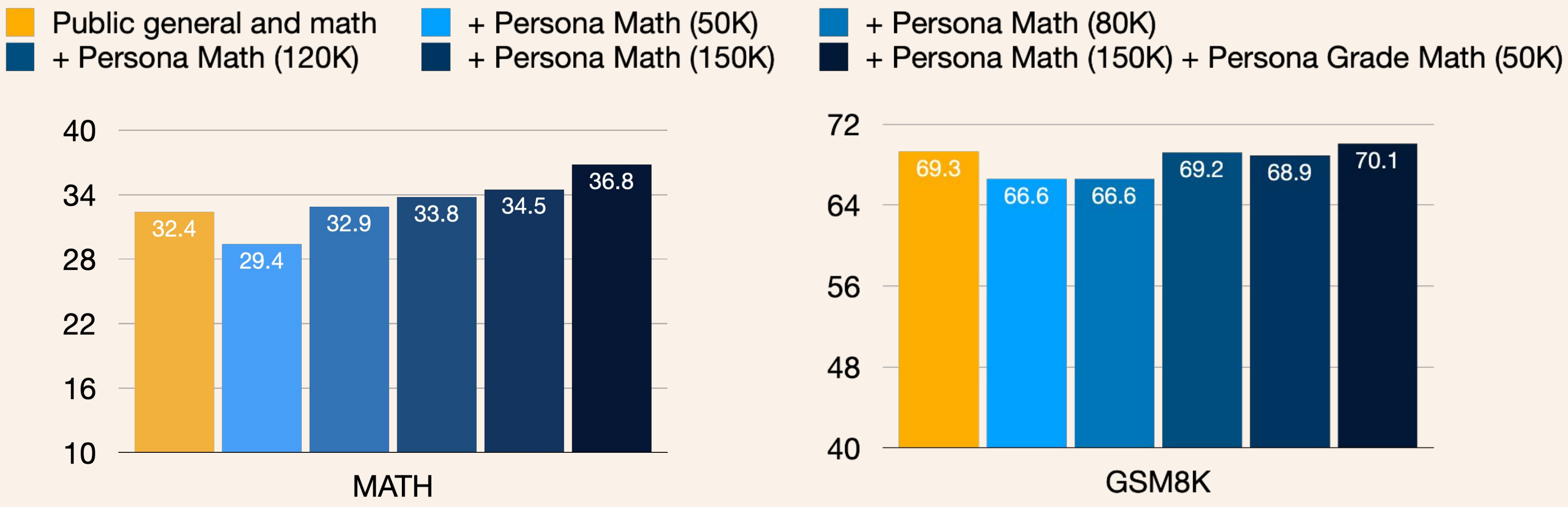


Impact of Persona-Driven Math Data

public datasets ⊕ X % **persona-driven synthetic math problems**

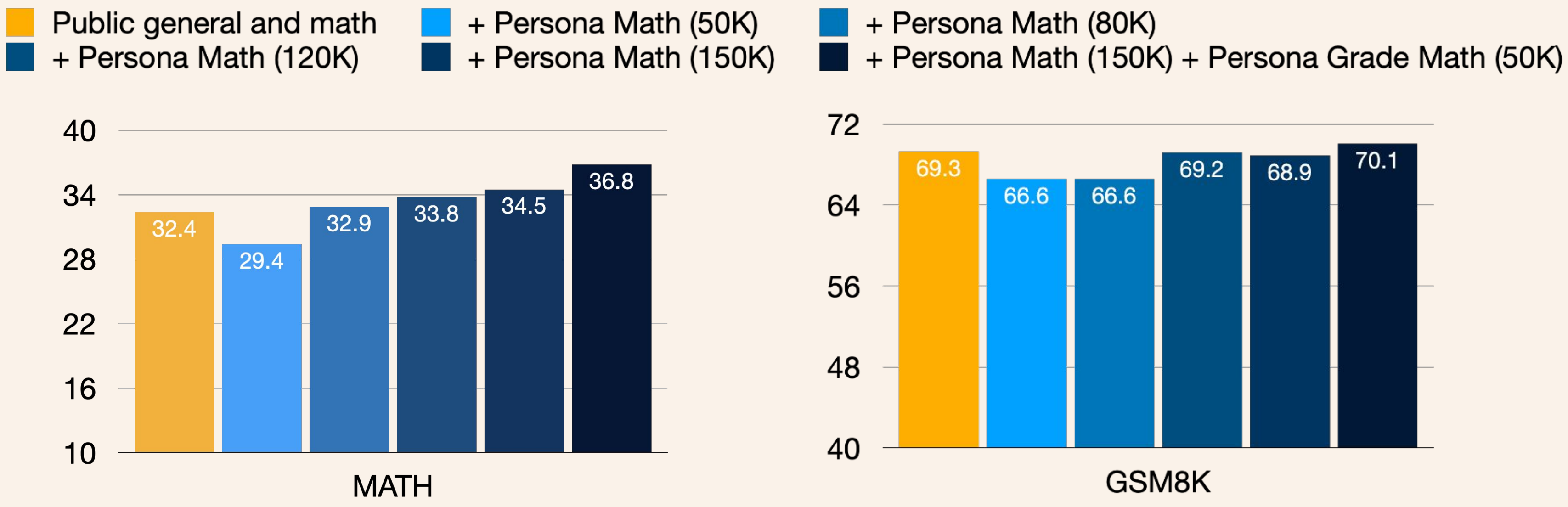
- General purpose (50K)
- NuminaMath-TIR (~64K)

- Hard math problems (150K)
- Grade school math (~50K)



Impact of Persona-Driven Math Data

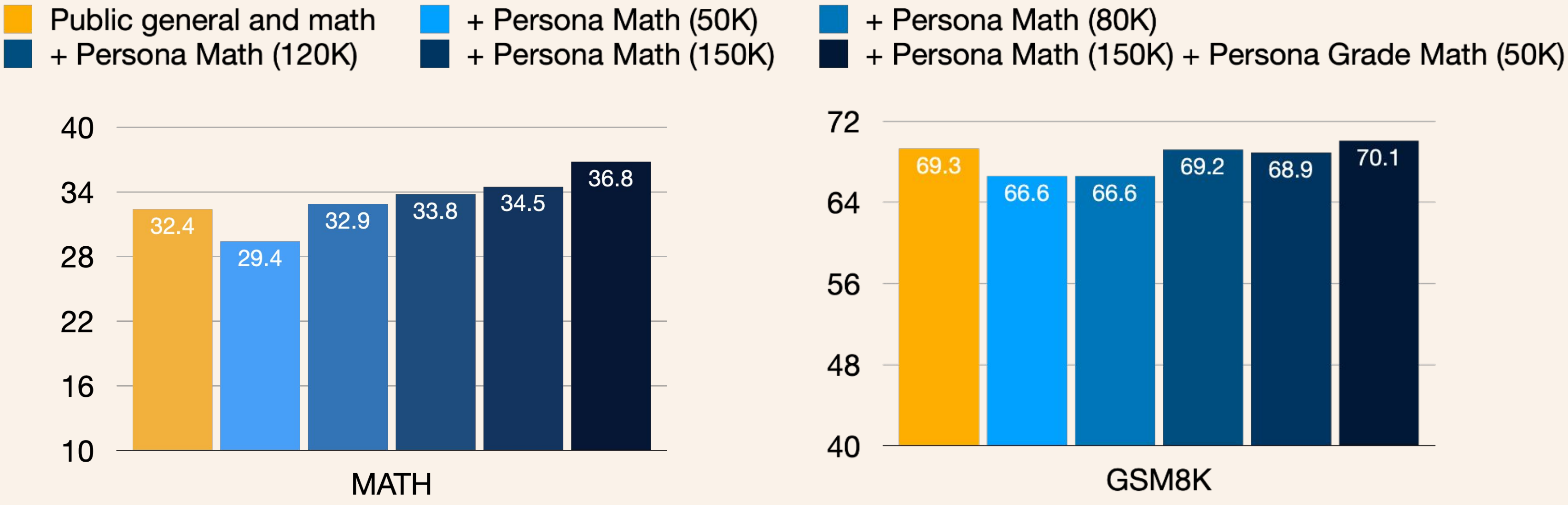
Adding more persona-driven math data, consistently improve MATH performance



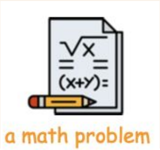
Impact of Persona-Driven Math Data

Adding more persona-driven math data, consistently improve MATH performance

- GSM8k improves (less than math)
- Adding grade-school math helps

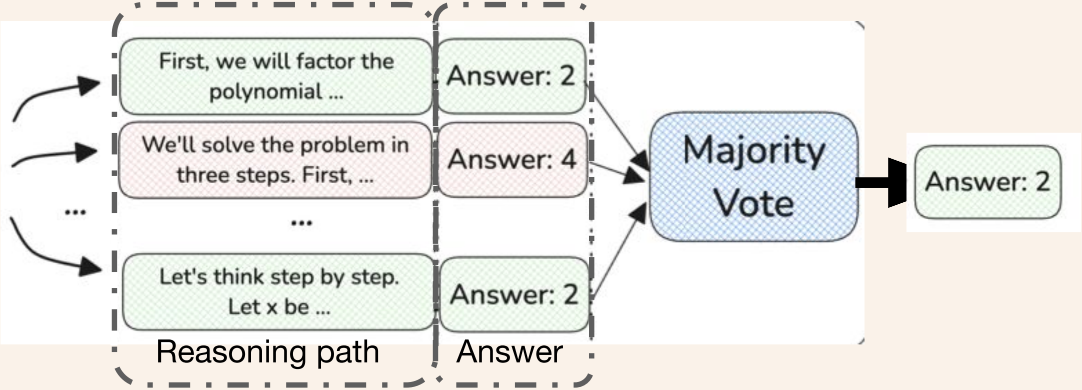


Improving data quality via voting / self-consistency



Dr. Smith, a chemist, is studying a reaction where compound X decomposes into products Y and Z. The reaction follows first-order kinetics with a rate constant k of 0.5 min^{-1}.

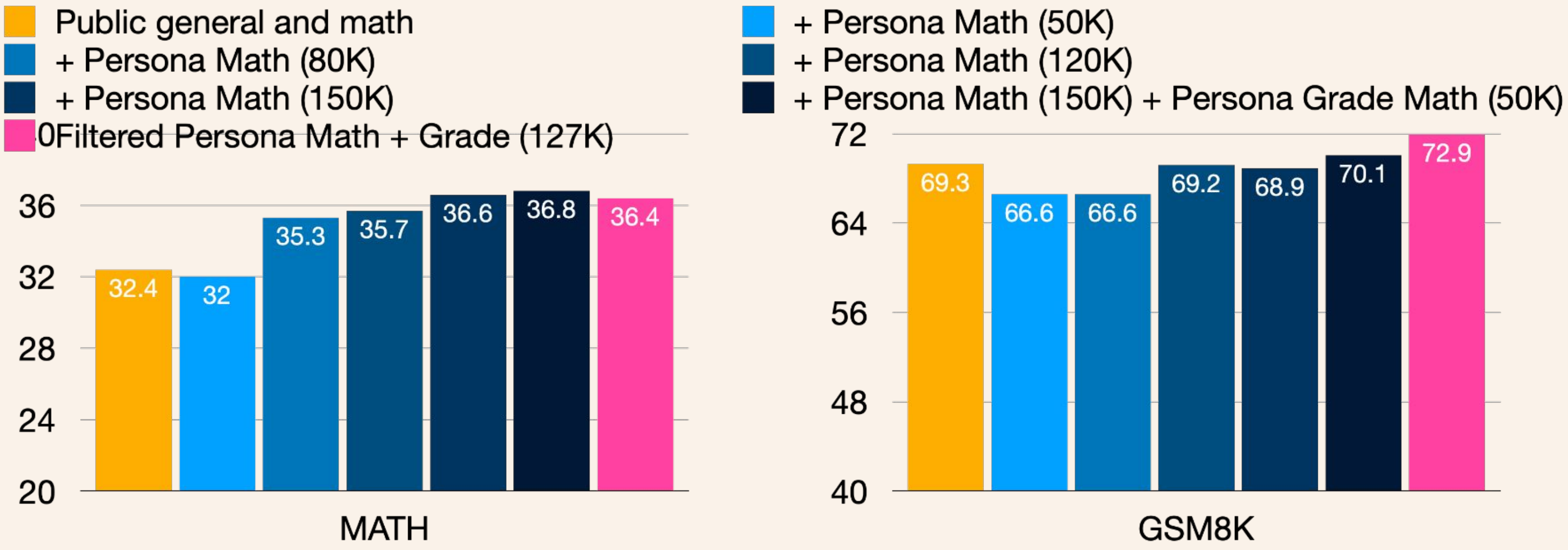
If the initial concentration of compound X is 1.0 M, how long will it take for the concentration of X to decrease to 0.25 M?



Remove instances with no majority vote!

Less data, Same or Better Performance

Using only ~60% of the data, we are still able to main the performance in MATH and improve in GSM8K



Other approaches to generate COT data

1. Manual Human Annotation (e.g., GSM8K dataset): Annotators write step by step solutions
   - High-quality reasoning traces
   - Limited **scale** (only 7K)
   - Lack of **diversity** in reasoning styles
2. Program-Aided Language Models (PAL): Convert math problems into Python code execution traces
   - Guarantee correctness through execution
   - Less natural language reasoning, less intuitive
   - Limited to problems that can be coded
3. Self-generated COT (self-ask): using LLMs to generate their reasoning paths
   - Scalable to many problems
   - **Quality** highly dependent on base model

Capability-driven data mixing

Core capabilities

- Chat
- Knowledge
- Reasoning
- Coding
- Safety
- Multilinguality
- Math
- Precise instruction following
- ...

→ Better

Human + Synthetic

data mix → Tulu SFT

Data mixing for SFT

Model	Avg.	MMLU	TQA	PopQA	BBH	CHE	CHE+	GSM	DROP	MATH	IFEval	AE 2	Safety
Tülu 3 8B SFT	**60.1**	62.1	46.8	29.3	67.9	**86.2**	**81.4**	76.2	61.3	31.5	**72.8**	12.4	93.1
→ w/o WildChat	58.9	61.0	45.2	28.9	65.6	85.3	80.7	75.8	59.3	31.8	70.1	7.5	**95.2**
→ w/o Safety	58.0	62.0	45.5	**29.5**	68.3	84.5	79.6	**76.9**	59.4	**32.6**	71.0	12.4	74.7
→ w/o Persona Data	58.6	**62.4**	**48.9**	29.4	68.3	84.5	79.0	76.8	**62.2**	30.1	53.6	**13.5**	93.9
→ w/o Math Data	58.2	62.2	47.1	**29.5**	**68.9**	86.0	80.5	64.1	60.9	23.5	70.6	12.0	93.5

Training on real user interactions with strong models is helpful almost across the board.

Safety training is largely orthogonal to the other skills.

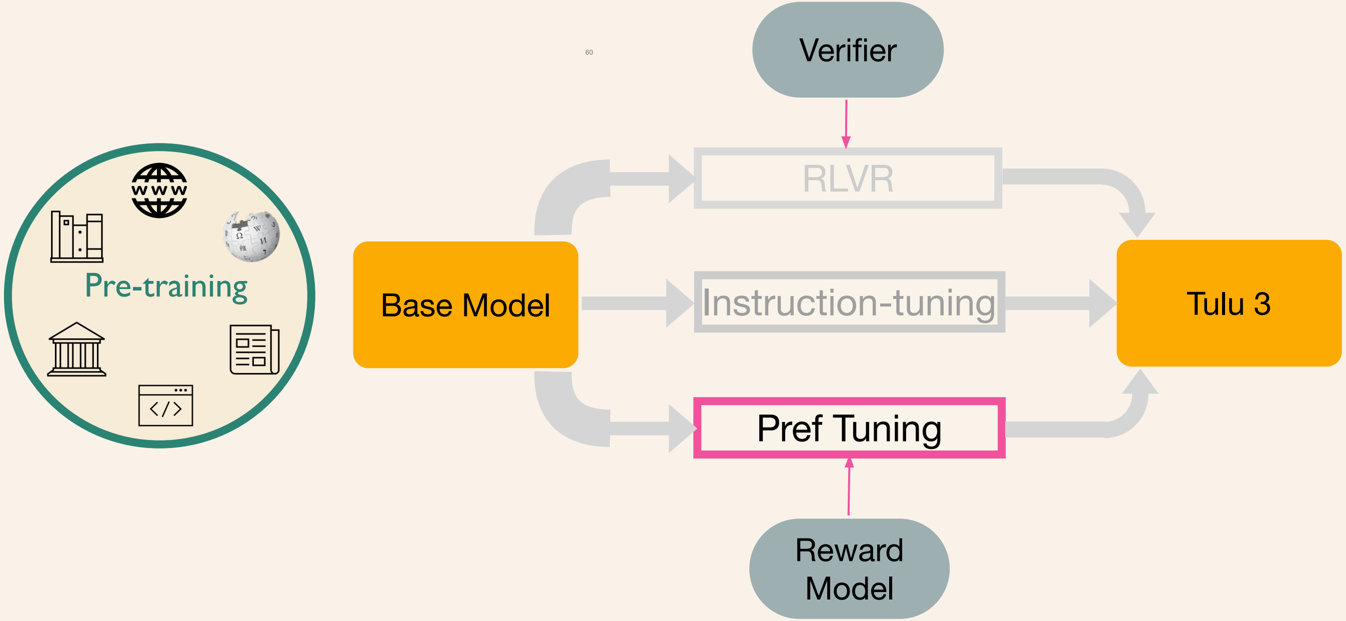
Persona-based data synthesis is very useful for targeting *new* skills.

SFT performance potential

Model	Avg.	MMLU	TQA	PopQA	BBH	CHE	CHE+	GSM	DROP	MATH	IFEval	AE 2	Safety
TÜLU 2 8B SFT	48.3	61.8	49.4	23.3	57.1	66.9	63.1	60.4	**61.7**	14.0	42.3	8.9	70.7
RLHFlow SFT V2	56.0	**65.8**	**56.0**	**29.7**	**69.3**	**86.2**	80.9	**81.6**	57.2	**35.7**	52.7	**13.6**	43.5
MAmmoTH2 8B	46.4	63.6	42.7	20.8	63.4	72.8	66.4	63.7	43.8	30.5	34.9	6.5	47.8
TÜLU 3 8B SFT	**60.1**	62.1	46.8	29.3	67.9	**86.2**	**81.4**	76.2	61.3	31.5	**72.8**	12.4	**93.1**
TÜLU 2 70B SFT	63.6	76.0	**57.8**	44.1	79.4	86.8	83.5	83.2	75.9	33.1	57.7	17.3	68.8
TÜLU 3 70B SFT	**72.6**	**79.4**	55.7	**48.6**	**82.7**	**92.9**	**87.3**	**91.1**	**77.2**	**53.7**	**82.1**	**26.3**	**94.4**

Table 8: Summary of the performance of our TÜLU 3 SFT models against comparable baselines. Our final SFT mixtures show strong performance, achieving a higher average score than other comparable mixes. All models, including TÜLU 2 SFT, were trained on either Llama 3.0 or 3.1. Our final Tülu 3 70B model was used to help format this table.

Tülu 3 Step 2: Preference tuning

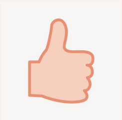


Step 2: Preference finetuning

Preference judgments

Input: `Write a haiku about AI`

Output 1: `Sure, here's a haiku: …`



Output 2: `Sorry, I cannot help you with that.`

Aligning to human preferences gives:

- Stronger training influence for style and chat evaluations (e.g. ChatBotArena).
- Continue building capabilities of skills from SFT, but lower absolute magnitude of improvements.

Step 2: Preference finetuning

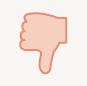
Aligning to human preferences gives:

- Stronger training influence for style and chat evaluations (e.g. ChatBotArena).
- Continue building capabilities of skills from SFT, but lower absolute magnitude of improvements.

Preference judgments

Input: `Write a haiku about AI`

Output 1: `Sure, here's a haiku: …` 

Output 2: `Sorry, I cannot help you with that.` 

Learning from human / AI feedback

Prompt : Explain the moon landing to a 6-year-old in a few sentences.

Response

In 1969, astronauts Neil Armstrong and Buzz Aldrin walked on the Moon for the first time, while Michael Collins stayed in the spaceship. They explored and collected moon rocks, showing the world that humans could visit another world in space.

Response

The moon landing was when some brave astronauts flew a special spaceship all the way to the moon! They drove a little car called the Lunar Module and walked on the moon's surface, leaving footprints in the dust. They even planted a flag and brought back some moon rocks to show everyone on Earth. It was a big adventure!

Preference instance

Learning from human / AI feedback

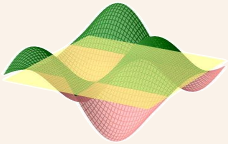
Prompt : Explain the moon landing to a 6-year-old in a few sentences.

Response		Response
In 1969, astronauts Neil Armstrong and Buzz Aldrin walked on the Moon for the first time, while Michael Collins stayed in the spaceship. They explored and collected moon rocks, showing the world that humans could visit another world in space.	<	The moon landing was when some brave astronauts flew a special spaceship all the way to the moon! They drove a little car called the Lunar Module and walked on the moon's surface, leaving footprints in the dust. They even planted a flag and brought back some moon rocks to show everyone on Earth. It was a big adventure!

Preference instance →

B is more engaging and suitable for 6-year-old

RLHF →

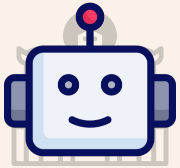


Learning from human / AI feedback

Prompt : Explain the moon landing to a 6-year-old in a few sentences.

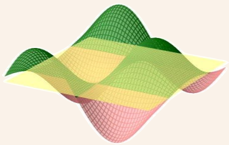
Response		Response
In 1969, astronauts Neil Armstrong and Buzz Aldrin walked on the Moon for the first time, while Michael Collins stayed in the spaceship. They explored and collected moon rocks, showing the world that humans could visit another world in space.	<	The moon landing was when some brave astronauts flew a special spaceship all the way to the moon! They drove a little car called the Lunar Module and walked on the moon's surface, leaving footprints in the dust. They even planted a flag and brought back some moon rocks to show everyone on Earth. It was a big adventure!

Preference instance →



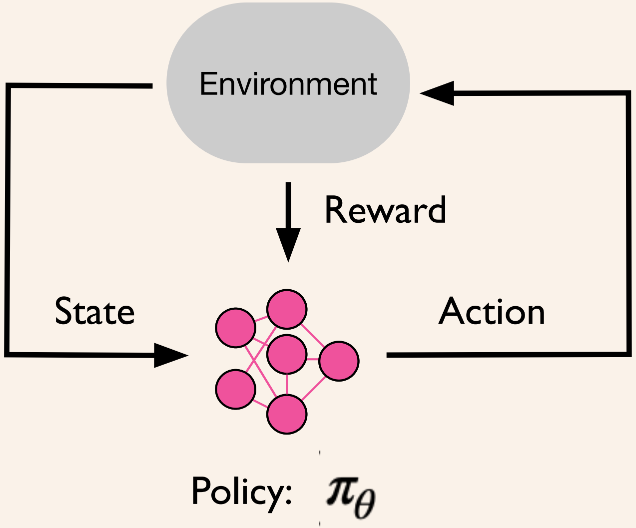
B is more engaging and suitable for 6-year-old

RLAIF →



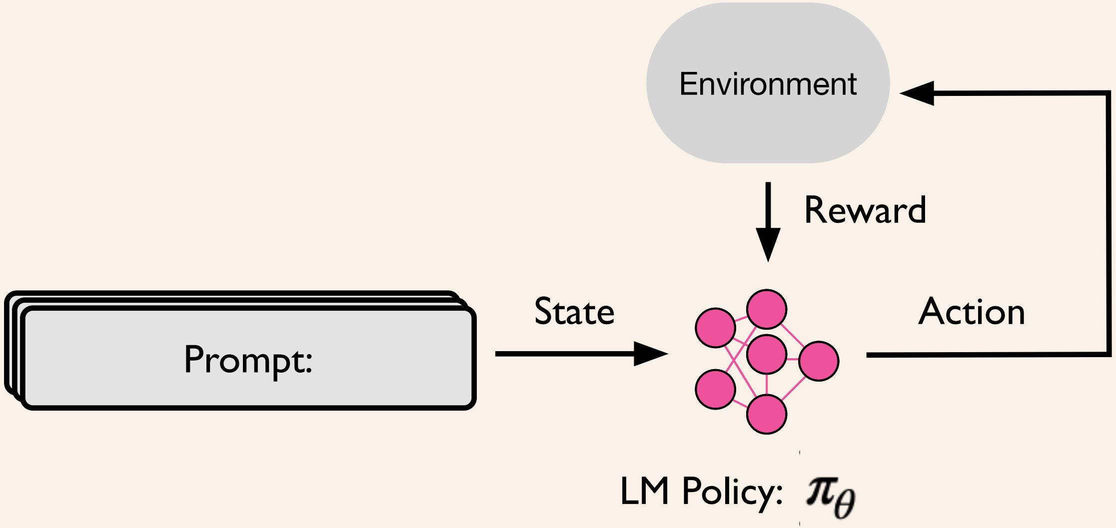
Step 2: Unpacking RLHF

Step 2: Unpacking RLHF

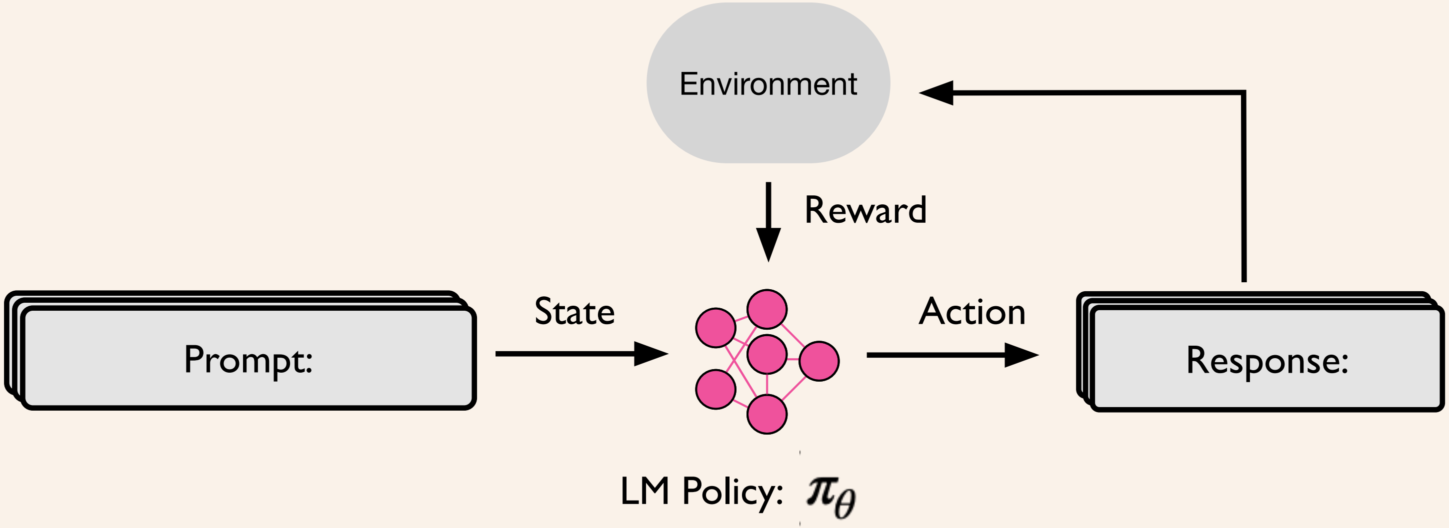


Ai2

Step 2: Unpacking RLHF

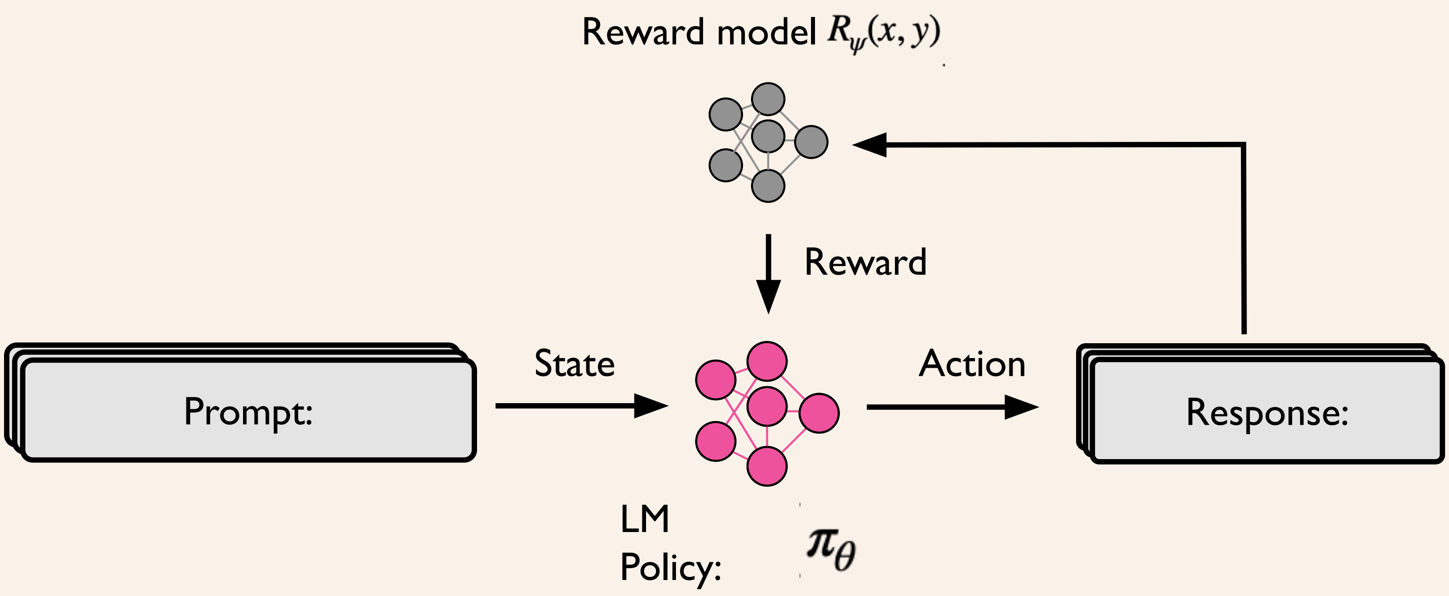


Step 2: Unpacking RLHF

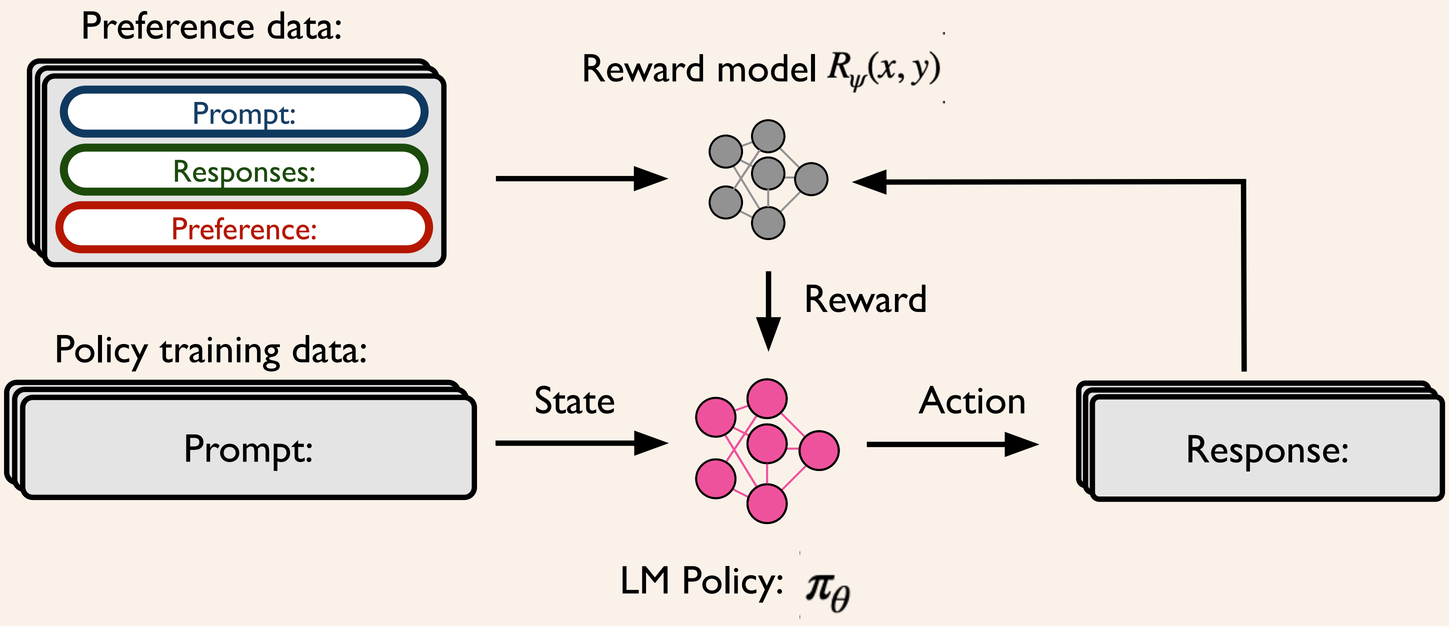


Ai2

Step 2: Unpacking RLHF

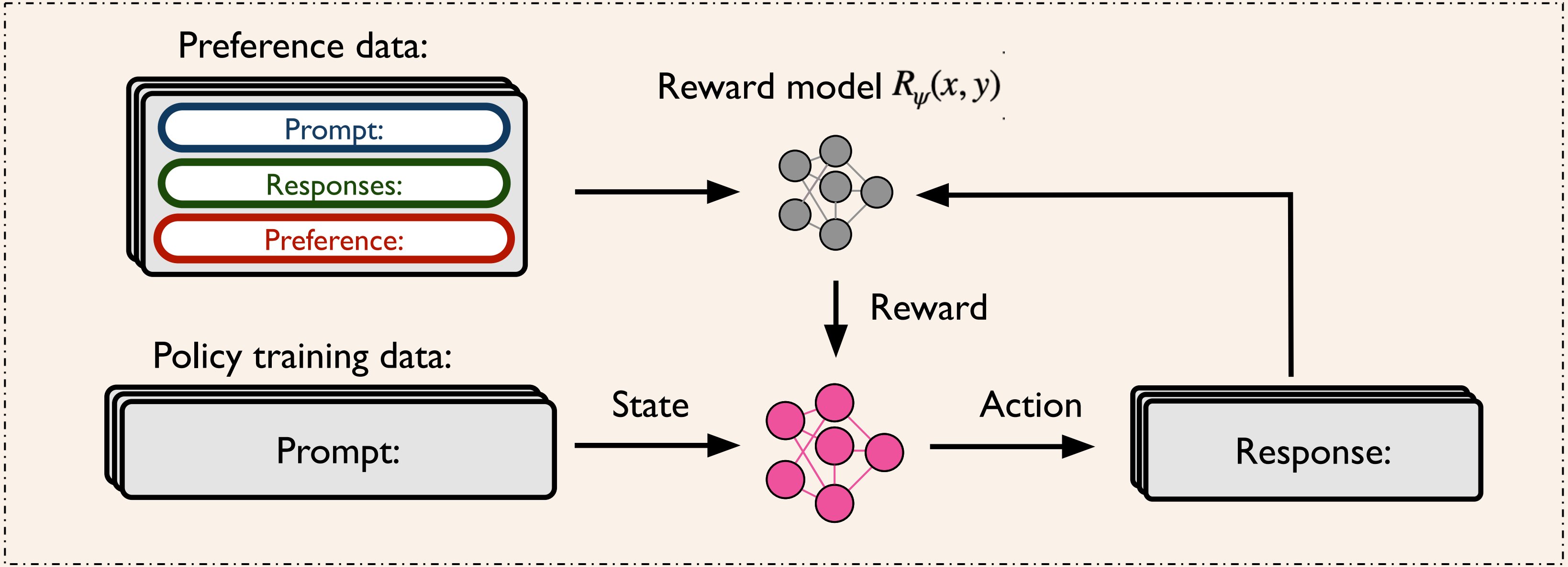


Step 2: Unpacking RLHF



[Christiano et al., 2017]

Step 2: Unpacking RLHF



PPO training:

$$\max_{\pi_\theta} \mathbb{E}_{x \sim \mathcal{D}_\pi, y \sim \pi_\theta(y|x)} \left[R_\psi(x, y) \right] - \beta \mathbb{D}_{\mathrm{KL}} \left(\pi_\theta \parallel \pi_{\mathrm{ref}} \right)$$

[Shulman et al., 2017]

RLHF objective → PPO

π: LLM policy
π_θ: base LLM
x: prompt
y: completion

$$\max_{\pi_\theta} \mathbb{E}_{x\sim\mathcal{D}, y\sim\pi_\theta(y|x)}\left[\underline{r_\phi(x,y)}\right] - \underline{\beta\mathbb{D}_{\mathrm{KL}}\left[\pi_\theta(y \mid x) \mid\mid \pi_{\mathrm{ref}}(y \mid x)\right]}$$

Optimize “reward” *inspired* ▲ by human preferences

▲ Constrain the model to stay close to the base LM (preferences are hard to model)

What if we just use gradient ascent on this equation?

$$\max_{\pi_\theta} \mathbb{E}_{x\sim\mathcal{D},y\sim\pi_\theta(y|x)}\left[r_\phi(x,y)\right] - \beta\mathbb{D}_{\mathrm{KL}}\left[\pi_\theta(y\mid x)\mid\mid\pi_{\mathrm{ref}}(y\mid x)\right]$$

The answer, with some math, is:

Direct Preference Optimization (DPO)

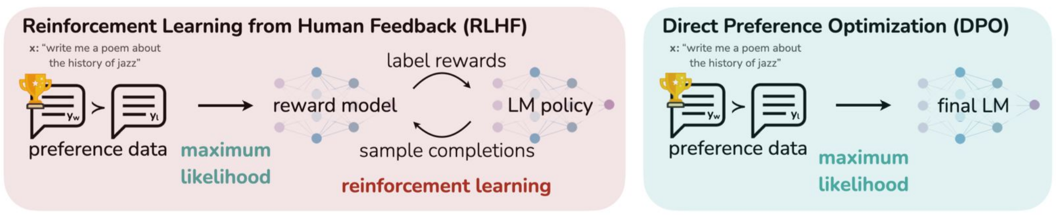


Figure 1: **DPO optimizes for human preferences while avoiding reinforcement learning.** Existing methods for fine-tuning language models with human feedback first fit a reward model to a dataset of prompts and human preferences over pairs of responses, and then use RL to find a policy that maximizes the learned reward. In contrast, DPO directly optimizes for the policy best satisfying the preferences with a simple classification objective, fitting an *implicit* reward model whose corresponding optimal policy can be extracted in closed form.

arXiv:2305.18290v2 [cs.LG] 13 Dec 2023

Direct Preference Optimization: Your Language Model is Secretly a Reward Model

Rafael Rafailov*† Archit Sharma*† Eric Mitchell*†
Stefano Ermon†‡ Christopher D. Manning† Chelsea Finn†
†Stanford University ‡CZ Biohub
{rafailov,architsh,eric.mitchell}@cs.stanford.edu

Abstract

While large-scale unsupervised language models (LMs) learn broad world knowledge and some reasoning skills, achieving precise control of their behavior is difficult due to the completely unsupervised nature of their training. Existing methods for gaining such steerability collect human labels of the relative quality of model generations and fine-tune the unsupervised LM to align with these preferences, often with reinforcement learning from human feedback (RLHF). However, RLHF is a complex and often unstable procedure, first fitting a reward model that reflects the human preferences, and then fine-tuning the large unsupervised LM using reinforcement learning to maximize this estimated reward without drifting too far from the original model. In this paper we introduce a new parameterization of the reward model in RLHF that enables extraction of the corresponding optimal policy in closed form, allowing us to solve the standard RLHF problem with only a simple classification loss. The resulting algorithm, which we call *Direct Preference Optimization* (DPO), is stable, performant, and computationally lightweight, eliminating the need for sampling from the LM during fine-tuning or performing significant hyperparameter tuning. Our experiments show that DPO can fine-tune LMs to align with human preferences as well as or better than existing methods. Notably, fine-tuning with DPO exceeds PPO-based RLHF in ability to control sentiment of generations, and matches or improves response quality in summarization and single-turn dialogue while being substantially simpler to implement and train.

1 Introduction

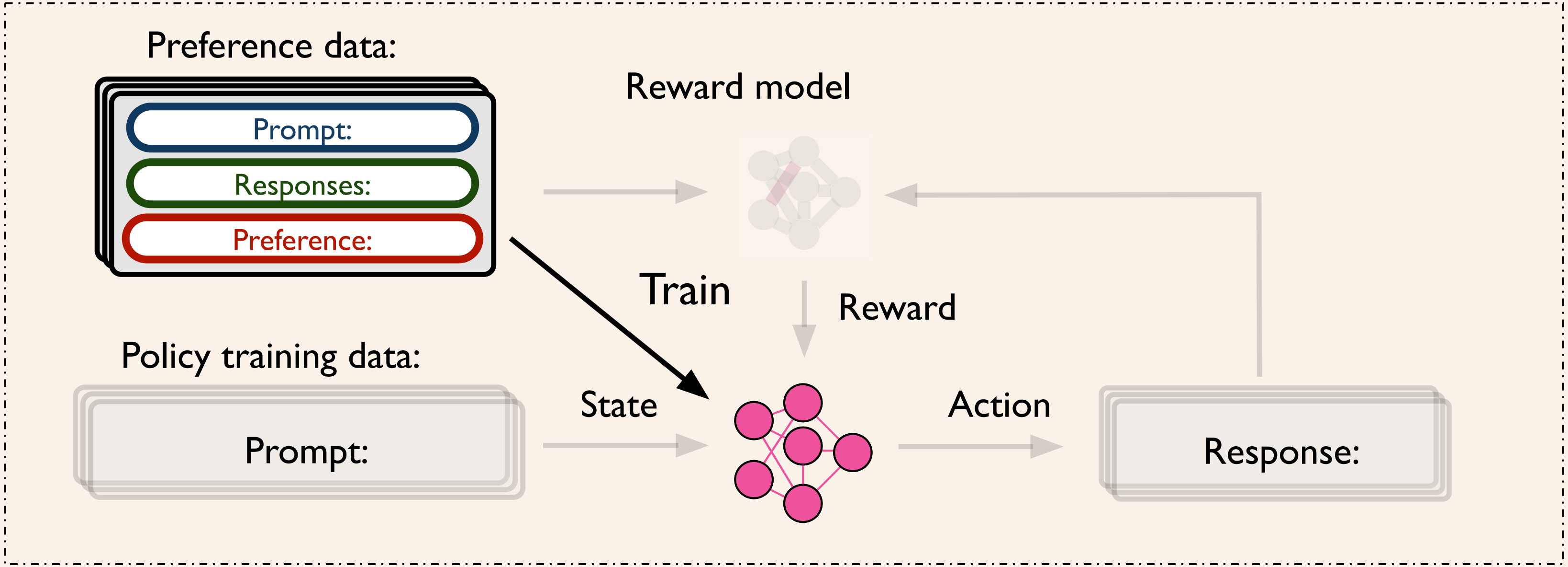
Large unsupervised language models (LMs) trained on very large datasets acquire surprising capabilities [11, 7, 40, 8]. However, these models are trained on data generated by humans with a wide variety of goals, priorities, and skillsets. Some of these goals and skillsets may not be desirable to imitate; for example, while we may want our AI coding assistant to *understand* common programming mistakes in order to correct them, nevertheless, when generating code, we would like to bias our model toward the (potentially rare) high-quality coding ability present in its training data. Similarly, we might want our language model to be *aware* of a common misconception believed by 50% of people, but we certainly do not want the model to claim this misconception to be true in 50% of queries about it! In other words, selecting the model's *desired responses and behavior* from its very wide *knowledge and abilities* is crucial to building AI systems that are safe, performant, and controllable [26]. While existing methods typically steer LMs to match human preferences using reinforcement learning (RL),

*Equal contribution; more junior authors listed earlier.

37th Conference on Neural Information Processing Systems (NeurIPS 2023).

Rafailov, Sharma, Mitchell et al. 2023

Step 2: Unpacking RLHF



DPO training:

$$\max_{\pi_\theta} \mathbb{E}_{(x,y_c,y_r)\sim\mathcal{D}_R}\left[\log \sigma\left(\beta \log \frac{\pi_\theta(y_c \mid x)}{\pi_{\text{ref}}(y_c \mid x)} - \beta \log \frac{\pi_\theta(y_r \mid x)}{\pi_{\text{ref}}(y_r \mid x)}\right)\right]$$

[Rafailov et al., 2023]

Ai2

Preference Tuning Optimization Algorithm

$$\max_{\pi_\theta} \mathbb{E}_{x\sim\mathcal{D}, y\sim\pi_\theta(y|x)}\left[r_\phi(x,y)\right] - \beta\mathbb{D}_{\mathrm{KL}}\left[\pi_\theta(y \mid x) \mid\mid \pi_{\mathrm{ref}}(y \mid x)\right]$$

Proximal Policy Optimization (PPO; Schulman et al., 2017) first trains a reward model and then uses RL to optimize the policy to maximize those rewards.

$$\mathcal{L}_{\mathrm{DPO}}(\pi_\theta;\pi_{\mathrm{ref}}) = -\mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}}\left[\log\sigma\left(\beta\log\frac{\pi_\theta(y_w \mid x)}{\pi_{\mathrm{ref}}(y_w \mid x)} - \beta\log\frac{\pi_\theta(y_l \mid x)}{\pi_{\mathrm{ref}}(y_l \mid x)}\right)\right]$$

Direct Preference Optimization (DPO; Rafailov et al., 2024) directly optimizes the policy on the preference dataset; no explicit reward model.

$$\mathcal{L}_{\mathrm{SimPO}}(\pi_\theta) = -\mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}}\left[\log\sigma\left(\frac{\beta}{|y_w|}\log\pi_\theta(y_w|x) - \frac{\beta}{|y_l|}\log\pi_\theta(y_l|x) - \gamma\right)\right]$$

SimPO (Meng et al., 2024) does not use a reference model.

Length-normalized DPO normalizes log-likelihoods of preferred and rejected responses by their lengths.

Preference Tuning Optimization Algorithm

PPO consistently outperforms DPO, but at the cost of:

- Implementation complexity
- Memory usage, and
- Throughput

Unpacking DPO and PPO: Disentangling Best Practices for Learning from Preference Feedback

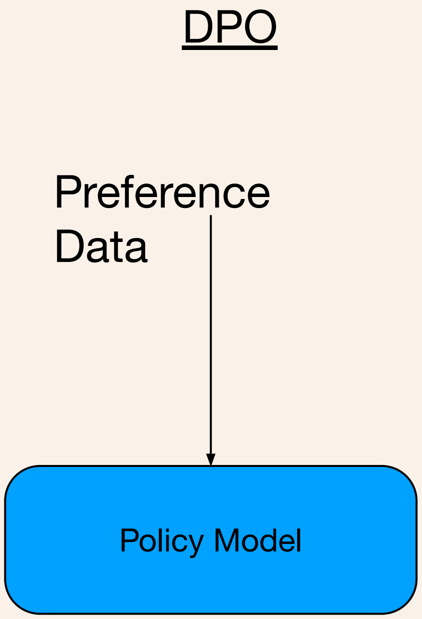
Hamish Ivison[♣♠] **Yizhong Wang**[♣♠] **Jiacheng Liu**[♣♠]
Zeqiu Wu[♠] **Valentina Pyatkin**[♣♠] **Nathan Lambert**[♣]
Noah A. Smith[♣♠] **Yejin Choi**[♣♠] **Hannaneh Hajishirzi**[♣♠]

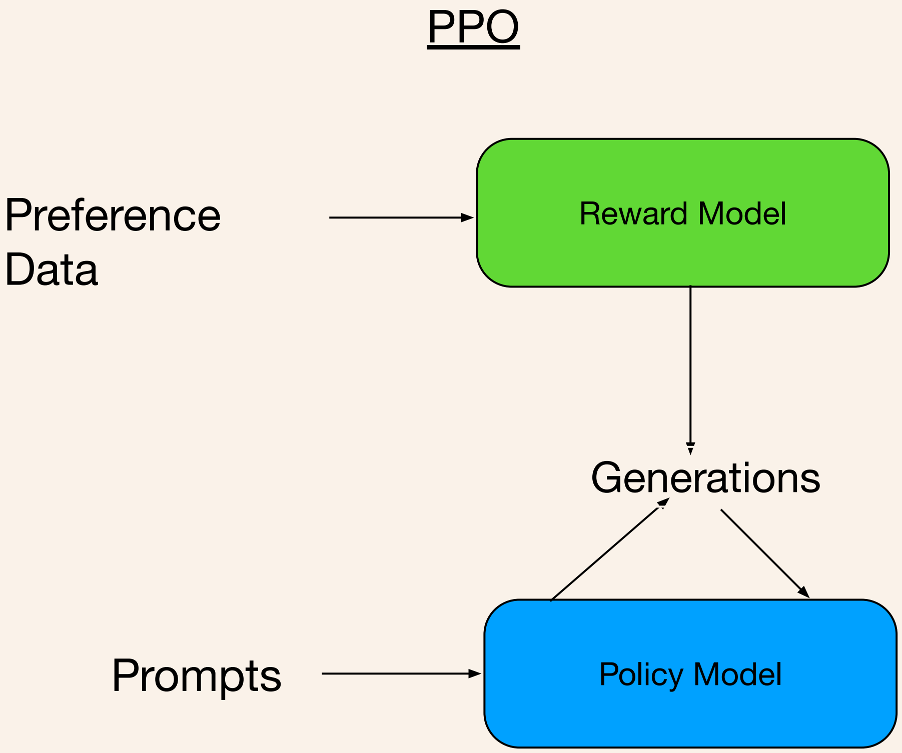
[♣]Allen Institute for AI [♠]University of Washington
hamishiv@cs.washington.edu

Normally can get ~1% improvement from switching from DPO to PPO

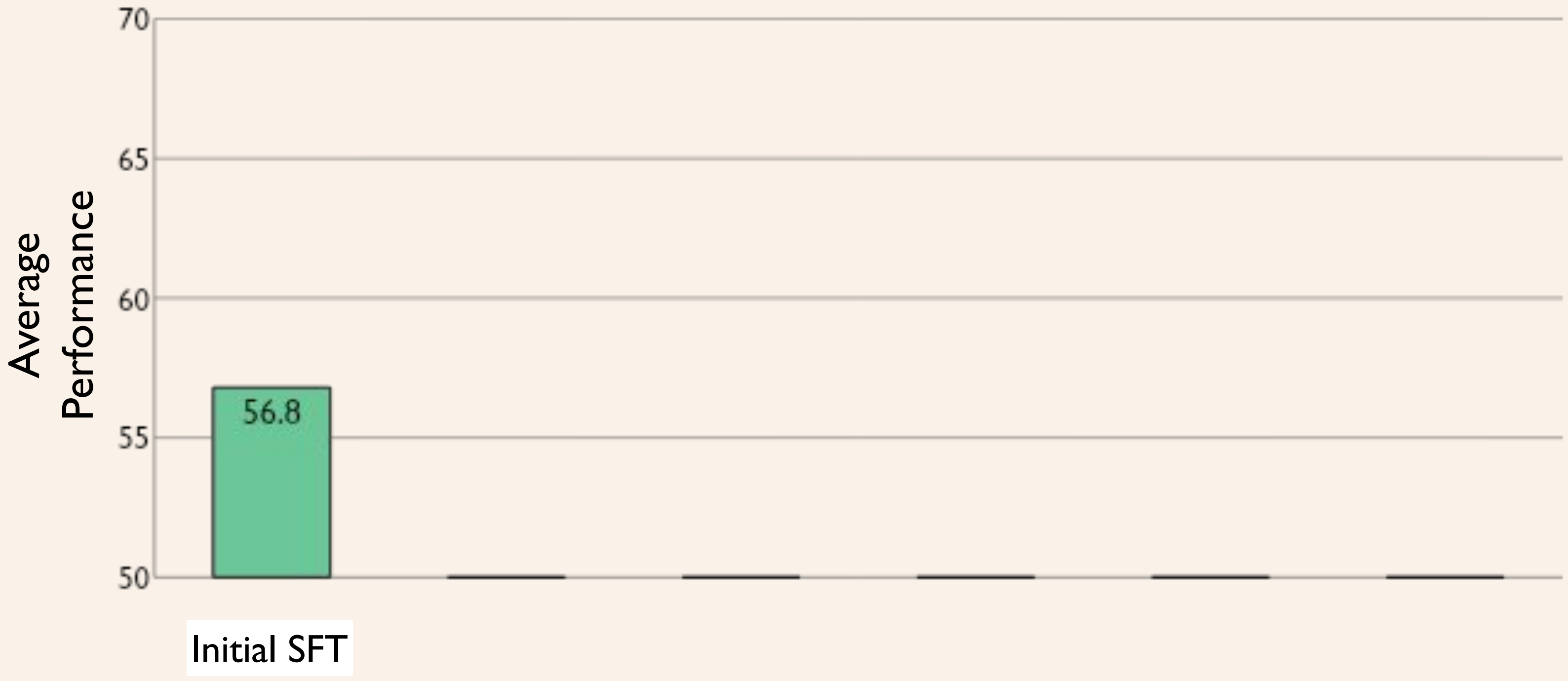
Ai2

DPO vs. PPO

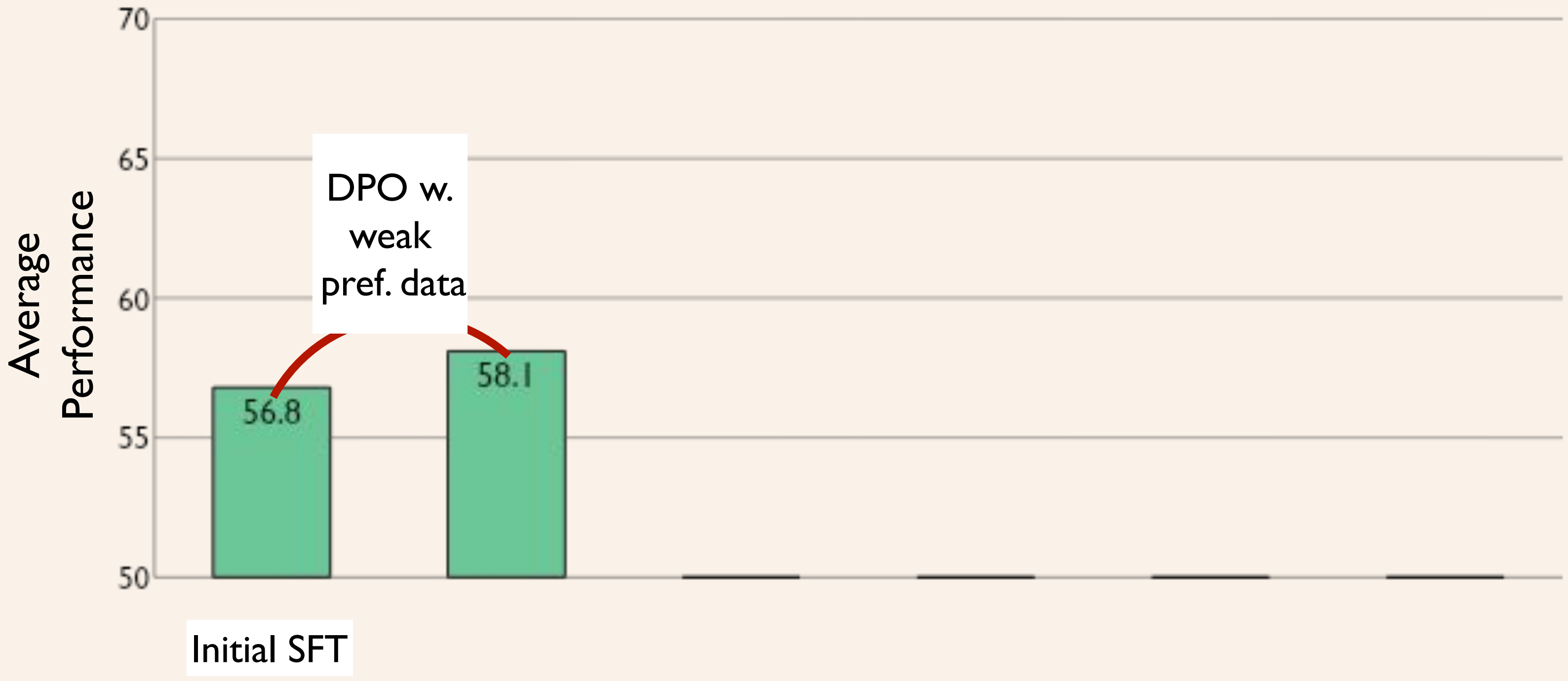

DPO
Preference Data
Policy Model


PPO
Preference Data
Reward Model
Generations
Prompts
Policy Model

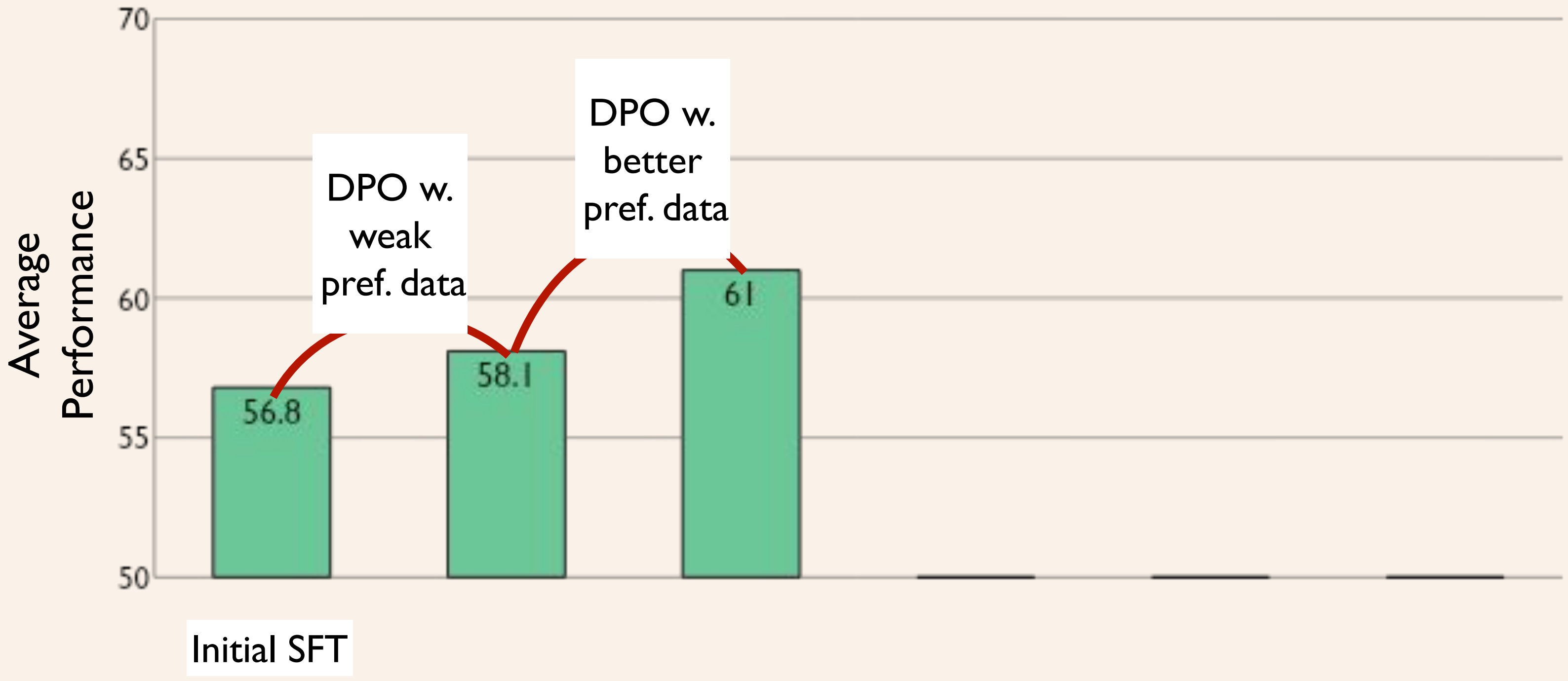
What components matter for LMs?



What components matter for LMs?

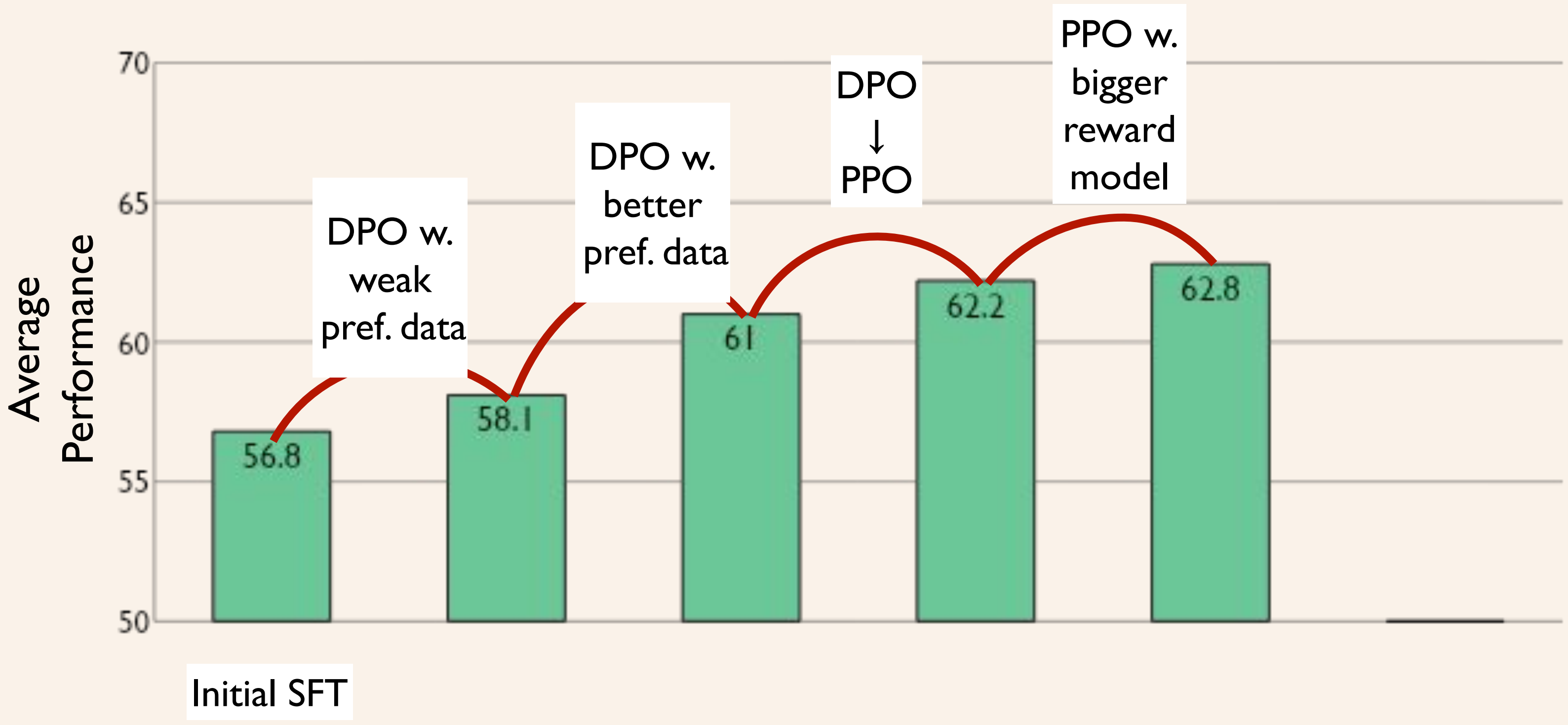


What components matter for LMs?



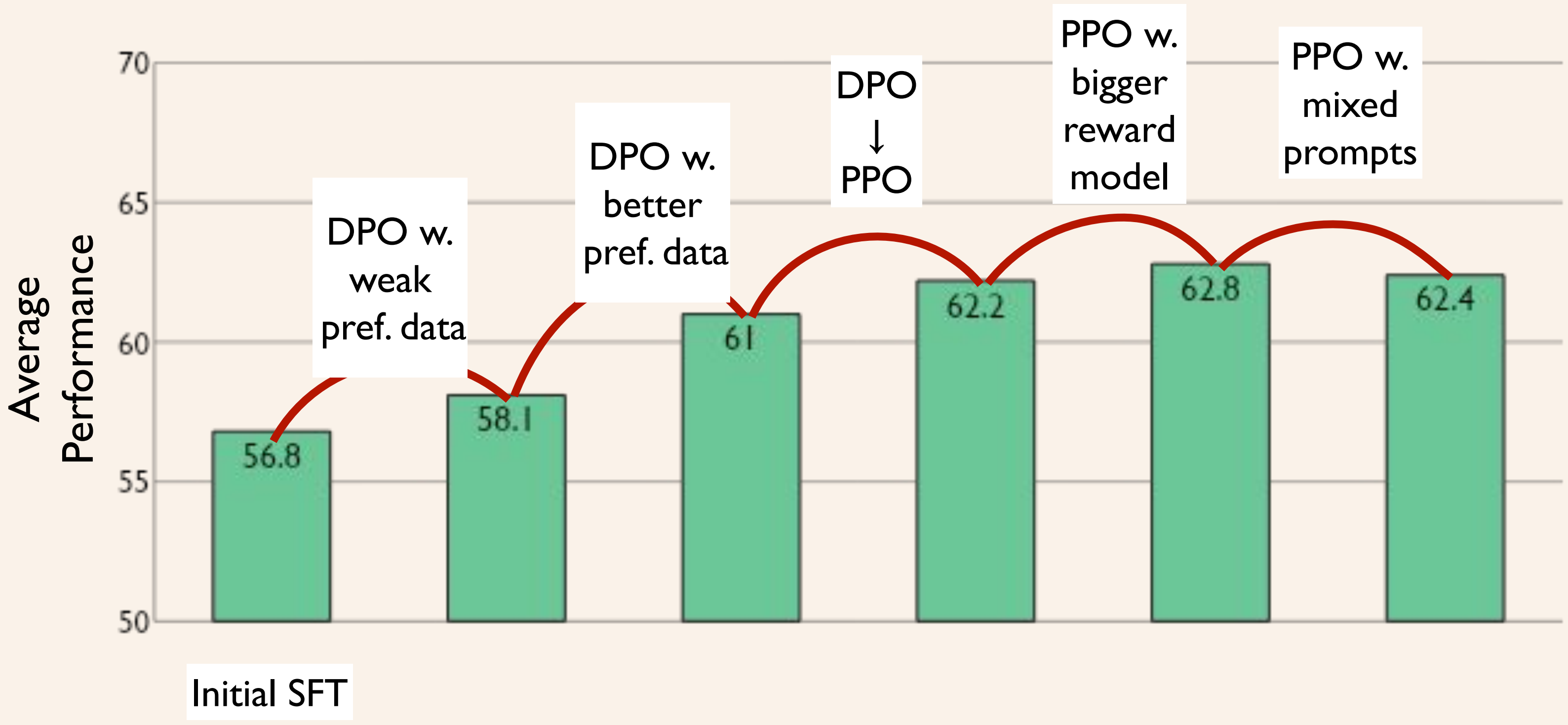
Ai2

What components matter for LMs?



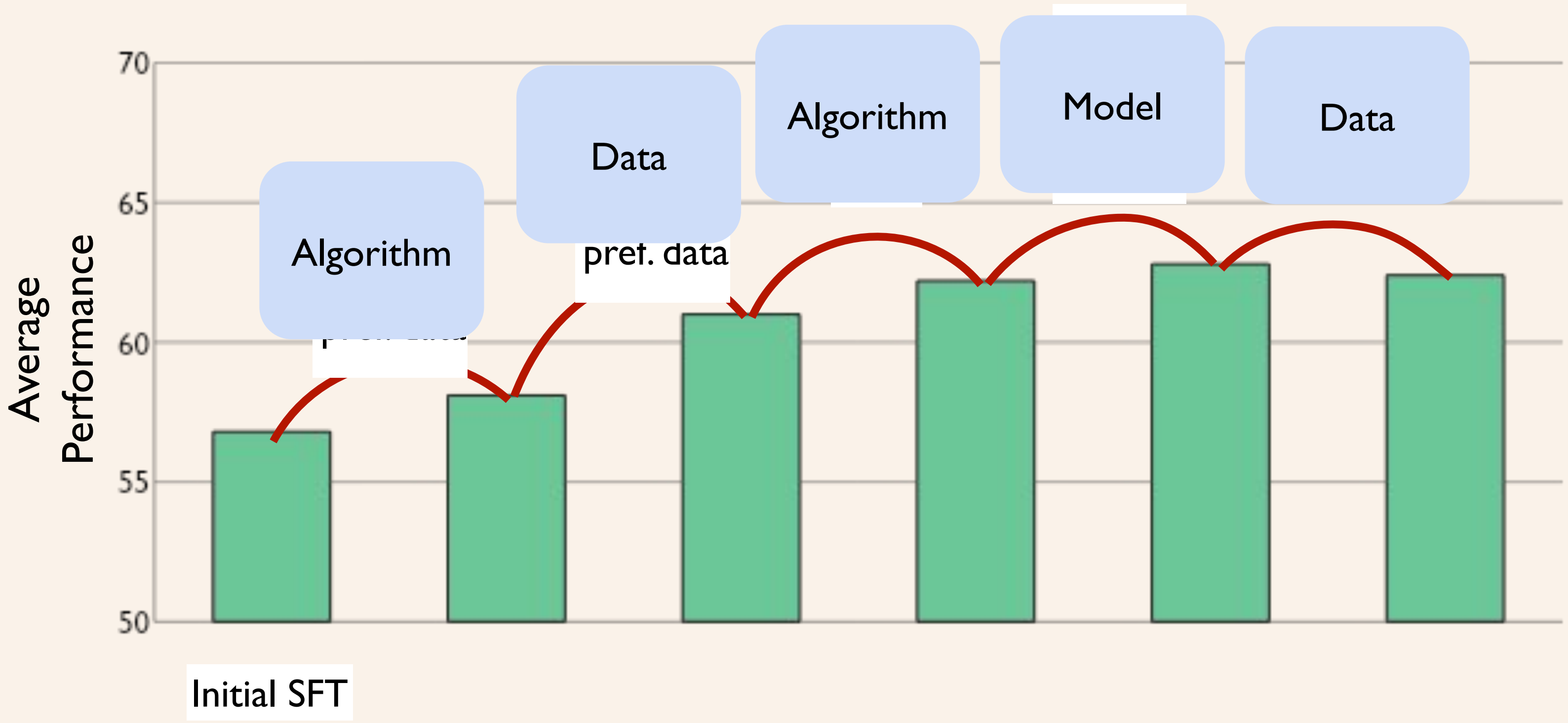
Ai2

What components matter for LMs?



Ai2

What components matter for LMs?

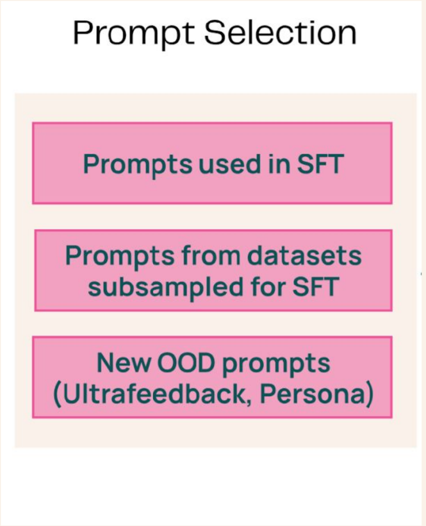


Ivison*, Wang* et al. 2023; Ivison, Wang et al. 2024

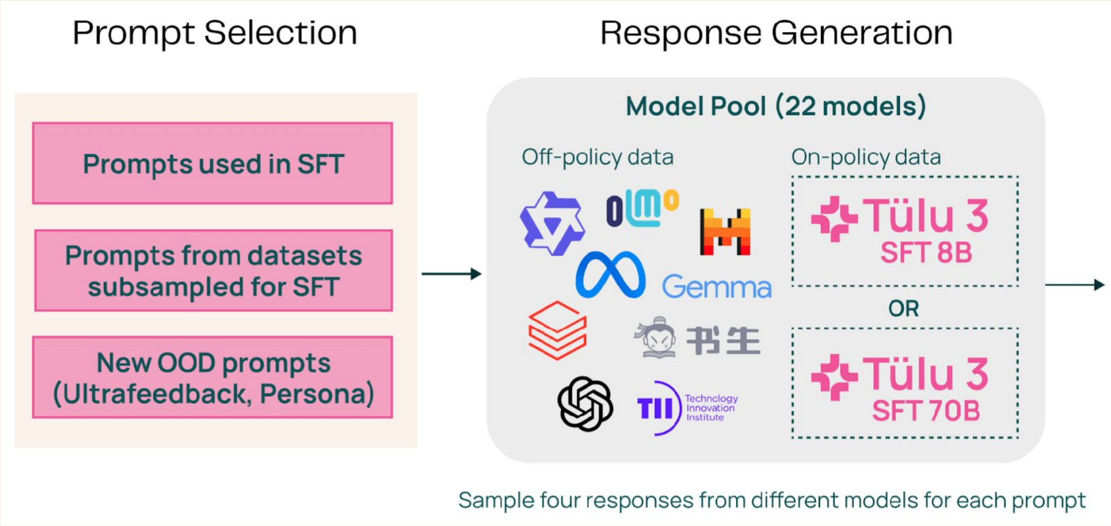
Takeaways

- Most important factor: High quality data
- PPO better than DPO in performance, but the cheapness of DPO makes it more practical for development
- Scaling RMs does not always yield better downstream models!
- Using in-domain prompts can yield further performance improvements

Putting all these for Tulu 3

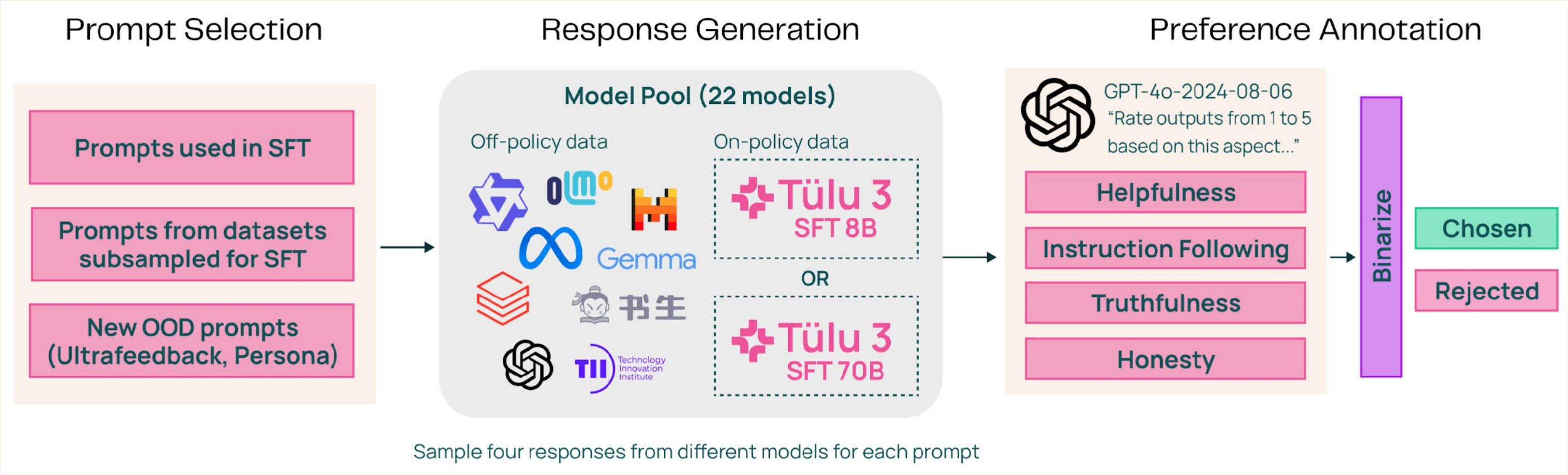


Putting all these for Tulu 3



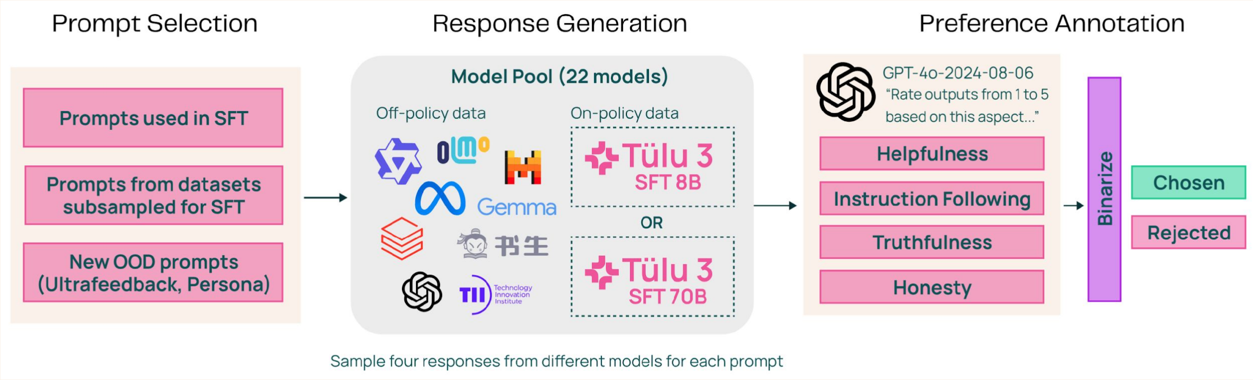
- We refined and scaled up the Ultrafeedback [Cui et al., 2023] for preference data generation.

Putting all these for Tulu 3

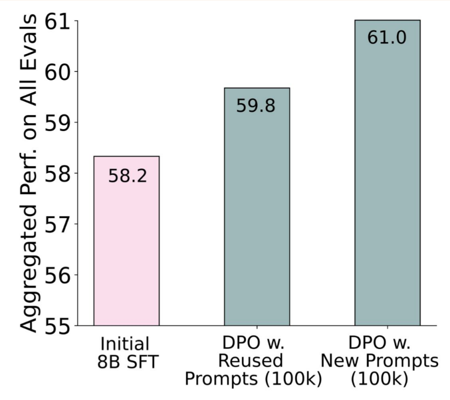


- We experimented with SimPO [Meng et al., 2024], but ended up with the length-normalized DPO.

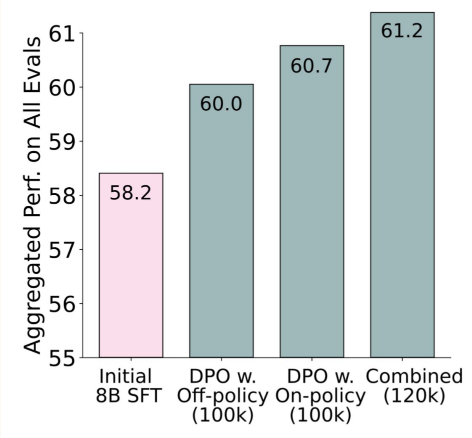
Step 2: Tulu 3 Preference tuning



Using SFT vs new prompts



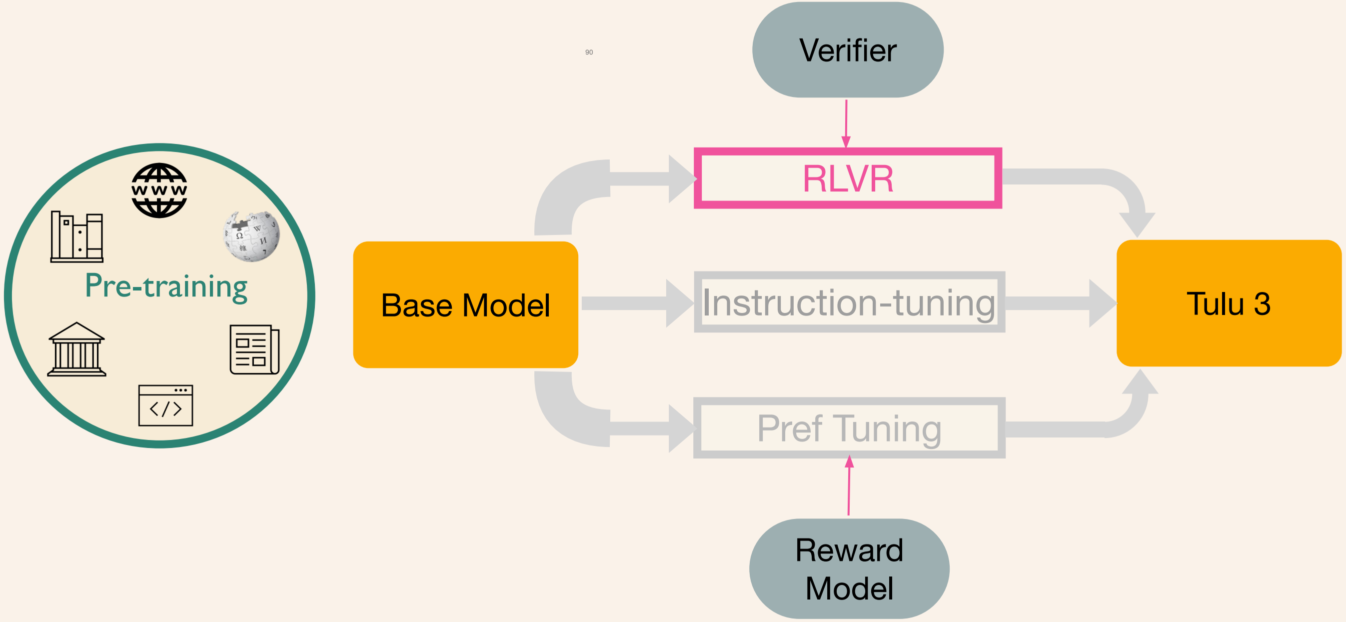
Off- vs On-policy preferences



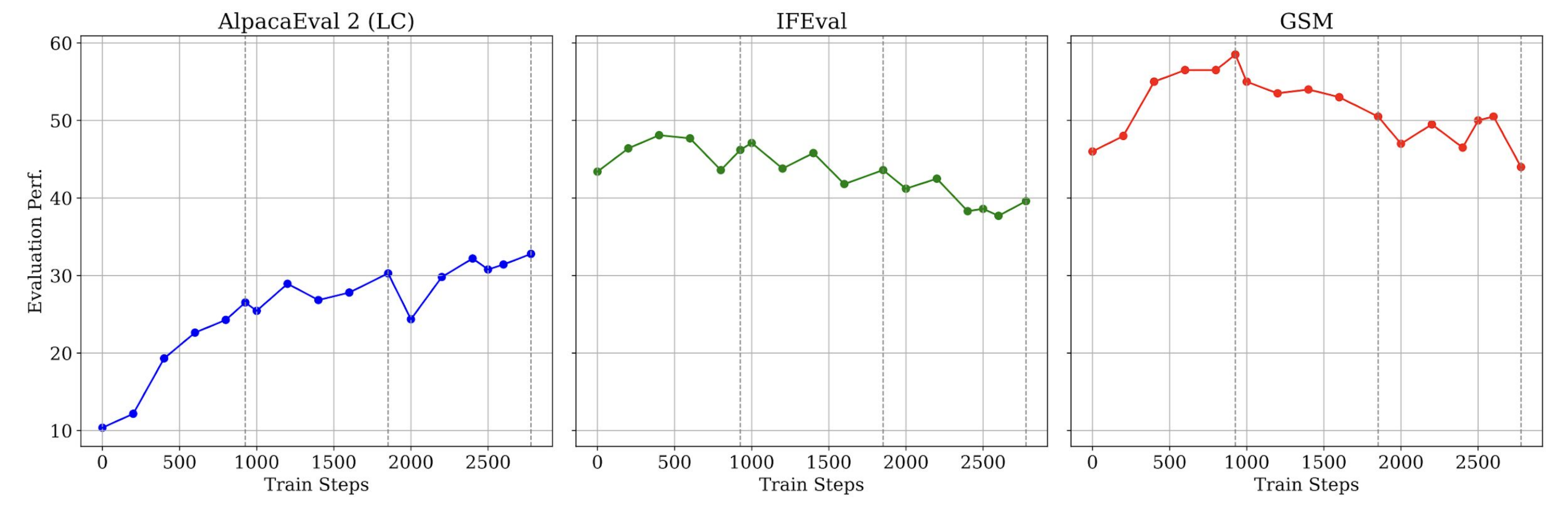
Different LM Judges

LLM Judge	Avg.
GPT-4o	57.3
LLama 3.1 405B	57.2
GPT-4 Turbo	57.0
GPT-4o Mini	56.9
Llama 3.1 70B	56.6

Tülu 3 Step 3: RLVR

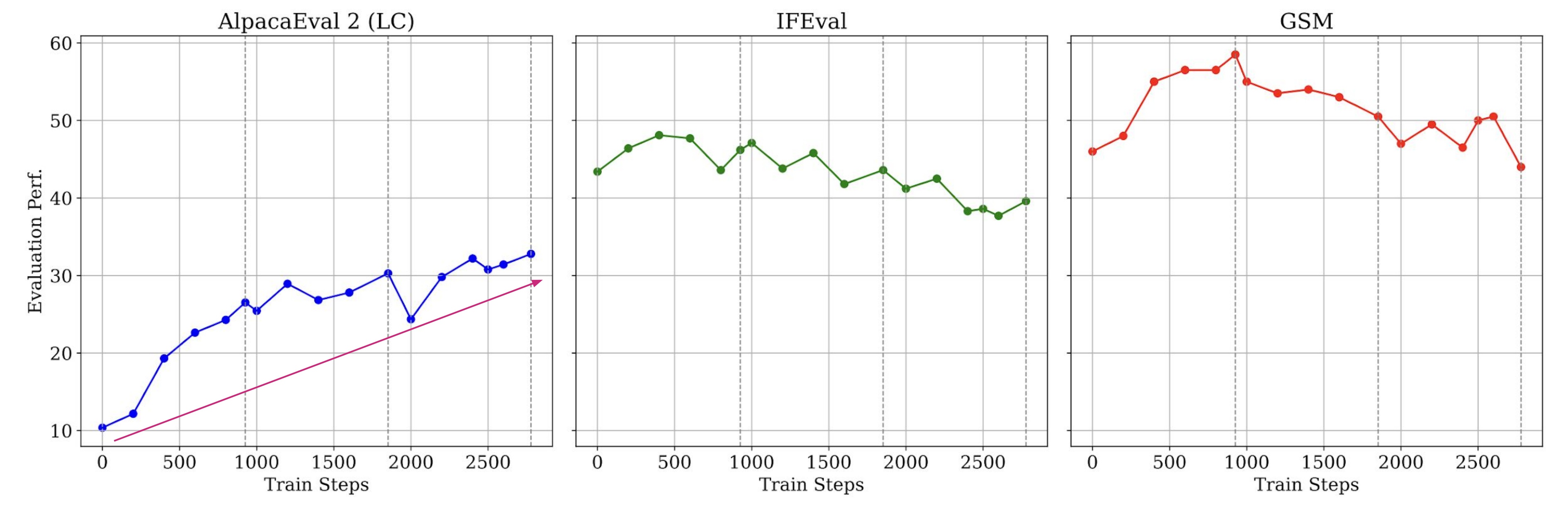


Over-optimization



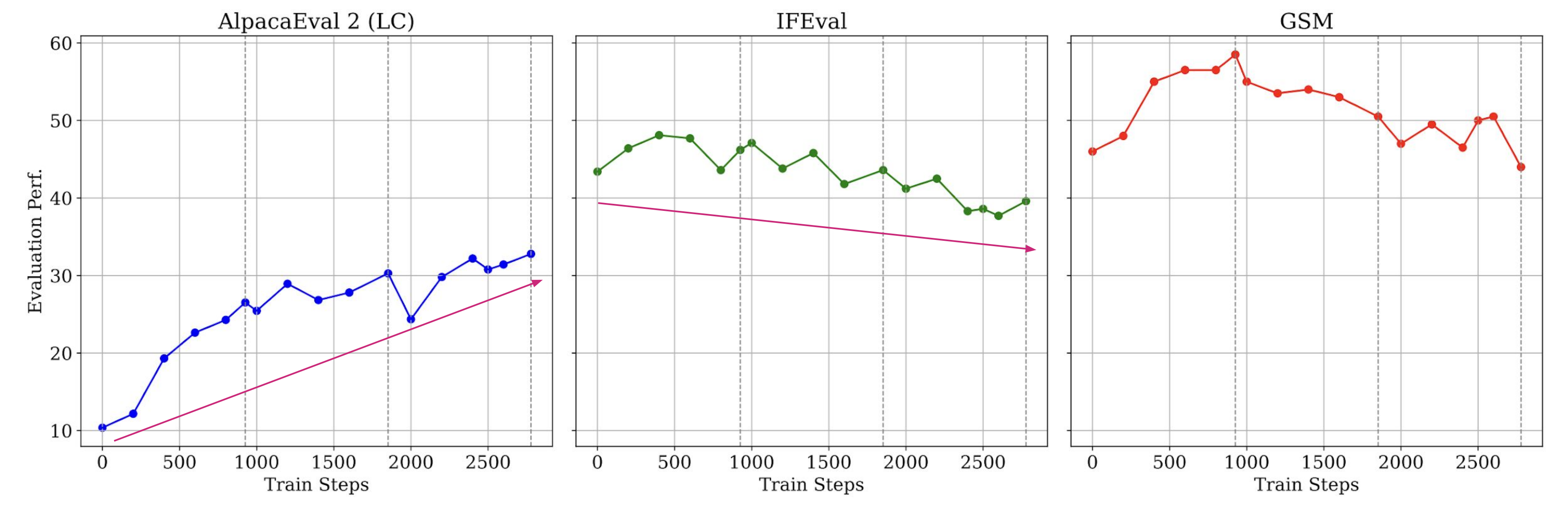
Ai2

Over-optimization



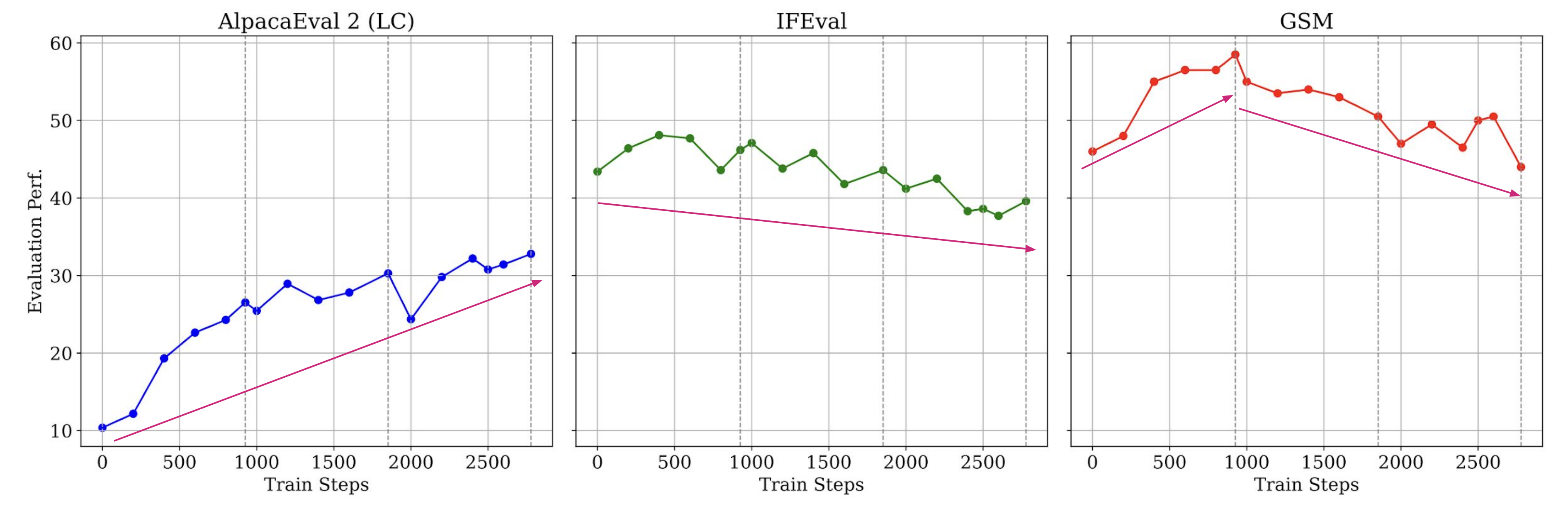
Ai2

Over-optimization



Ai2

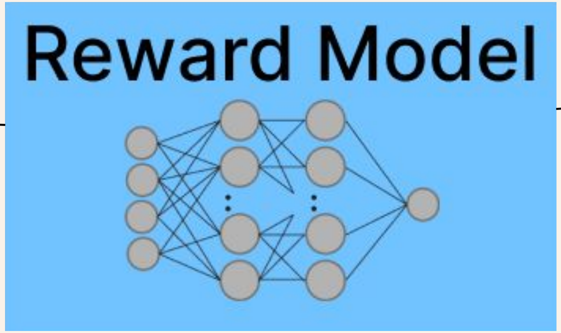
Perils of over-optimization



Ai2

Why? Neural RM…

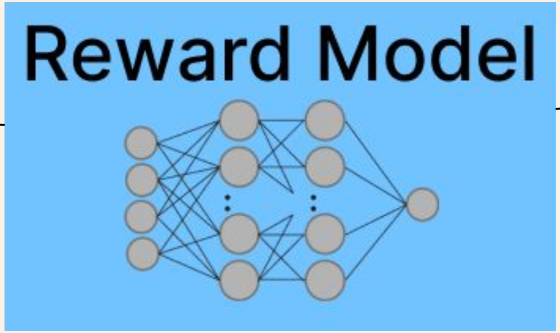
What is a Tulu? A Tulu is a camel that…



Score: 10.5

Why? Neural RM…

What is a Tulu? A Tulu is a camel that…



Score: 10.5

A Long Way to Go: Investigating Length Correlations in RLHF

Prasann Singhal
The University of Texas at Austin
prasanns@cs.utexas.edu

Tanya Goyal
Princeton University
tanyagoyal@princeton.edu

Jiacheng Xu
Salesforce AI
jiacheng.xu@salesforce.com

Greg Durrett
The University of Texas at Austin
gdurrett@cs.utexas.edu

HUMAN FEEDBACK IS NOT GOLD STANDARD

Tom Hosking
University of Edinburgh
tom.hosking@ed.ac.uk

Phil Blunsom
Cohere
phil@cohere.com

Max Bartolo
Cohere, UCL
max@cohere.com

Simplifying the reward model: rule-based rewards

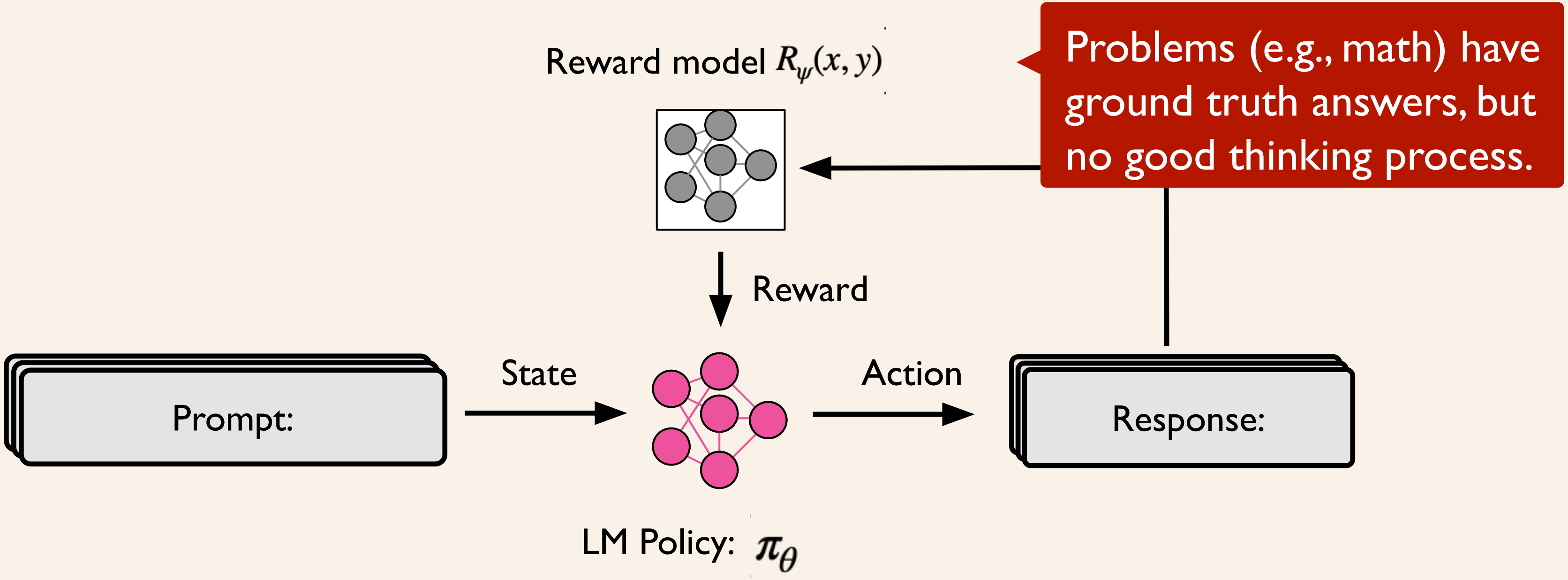
What is 2+2? 4.

```
if answer == gold label:
    return 1
else:
    return 0
```

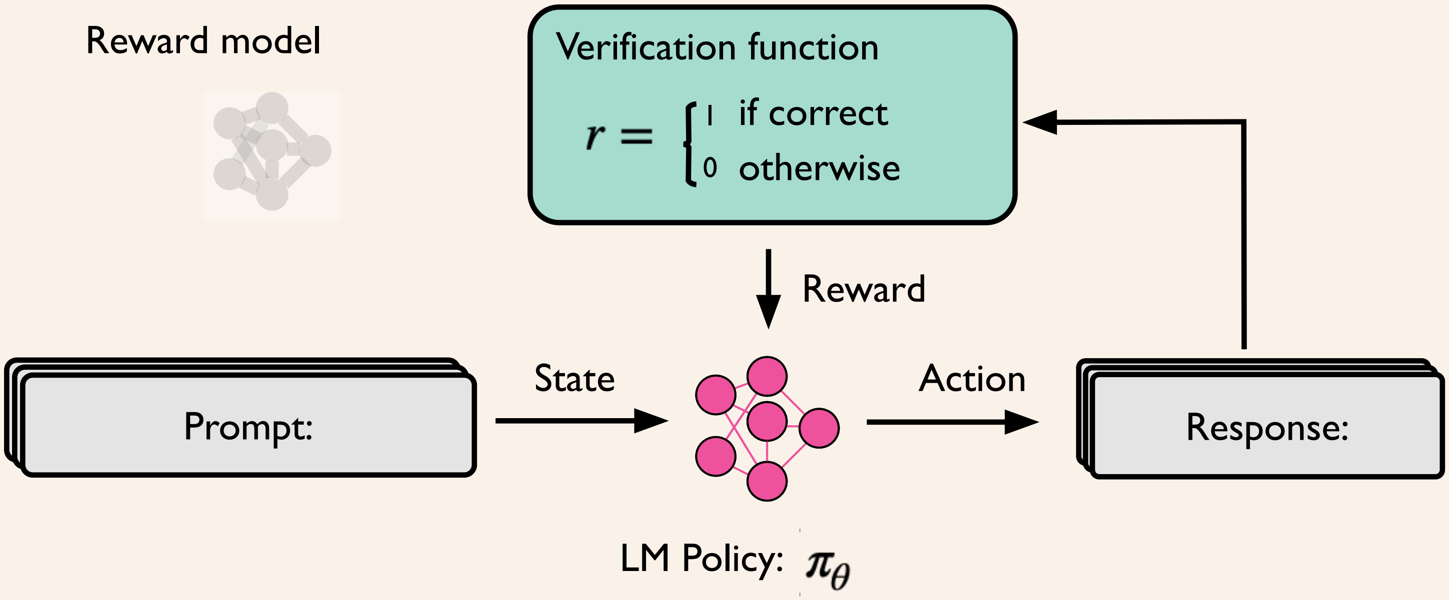
Score: 1

Can we just remove this complex setup and use simpler ‘models’…?

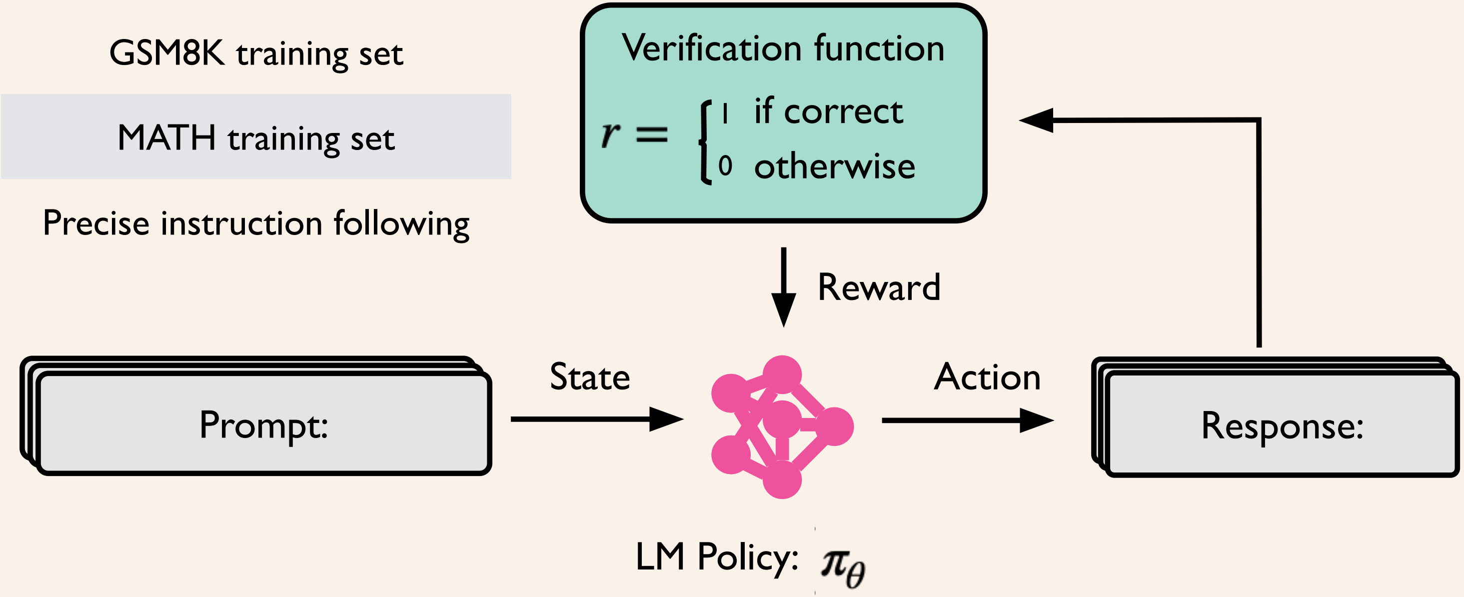
Tülu 3: RL with verifiable rewards



Tülu 3: RL with verifiable rewards



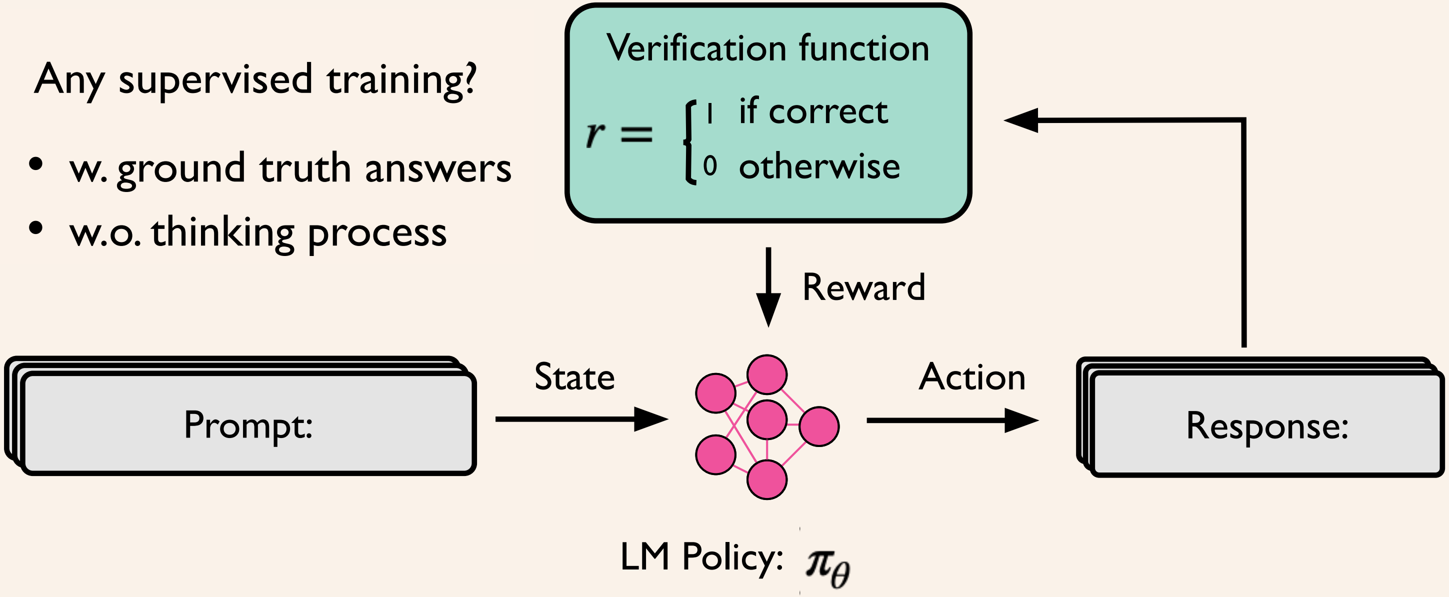
Tülu 3: RL with verifiable rewards



Tülu 3: RL with verifiable rewards

Any supervised training?

- w. ground truth answers
- w.o. thinking process



Ai2

Tülu 3: RL with verifiable rewards

December 6, 2024

OpenAI's Reinforcement Fine-Tuning Research Program

We're expanding our Reinforcement Fine-Tuning Research Program to enable developers and machine learning engineers to create expert models fine-tuned to excel at specific sets of complex, domain-specific tasks.

/erifi

deepseek R1

2.2.2. Reward Modeling

The reward is the source of the training signal, which decides the optimization direction of RL. To train DeepSeek-R1-Zero, we adopt a rule-based reward system that mainly consists of two types of rewards:

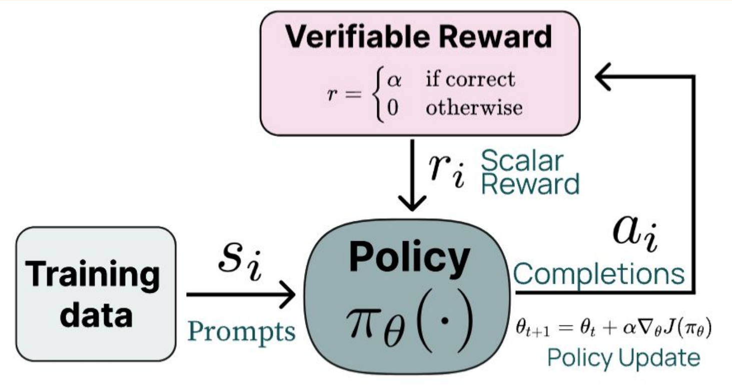
- **Accuracy rewards**: The accuracy reward model evaluates whether the response is correct. For example, in the case of math problems with deterministic results, the model is required to provide the final answer in a specified format (e.g., within a box), enabling reliable rule-based verification of correctness. Similarly, for LeetCode problems, a compiler can be used to generate feedback based on predefined test cases.
- **Format rewards**: In addition to the accuracy reward model, we employ a format reward model that enforces the model to put its thinking process between '<think>' and '</think>' tags.

LM Policy:

Lambert, …, Wang, Dasigi, Hajishirzi, 2024

Step 3: Reinforcement learning w. verifiable rewards

- ✅ Gold final answers or verifiable constraints.
- ❌ intermediate chain of thoughts or not matching model.
- Classical RL! (We used PPO for optimization)
- We tried it using three datasets.



Prompt Dataset	Count	Verification
GSM8K Train	7,473	Exact match against extracted answer
MATH Train	7,500	Exact match against extracted answer
IF verifiable	14,973	Prompt-specific verifiers
Total	29,946	

Ai2

Experimental Setup

1. Start from Tulu 3 DPO and SFT

2. Use a targeted dataset + paired verifier

3. Train with PPO

Evaluation	Training Data
GSM8k	GSM8k train set (~7k)
MATH	MATH train set (~7k)
IFEval	IF persona set(~15k)
BBH	Flan dataset (~90k)

Experimental Setup

1. Start from Tulu 3 DPO and SFT

2. Use a targeted dataset + paired verifier

3. Train with PPO

```python
def verify_gsm8k_sample(model_output, ground_truth_answer):
    # gsm is easy: extract numbers, and then just compare last number with answer.
    # matches how we do eval.
    predictions = None
    # replace numbers like `x,xxx` with `xxxx`
    response = re.sub(r"(\d),(\d)", r"\1\2", model_output)
    numbers = re.findall(r"[-+]?\d*\.\d+|\d+", response)
    if numbers:
        predictions = numbers[-1]
    else:
        predictions = response
    return str(predictions).lower() == str(ground_truth_answer).lower()
```

Experimental Setup

1. Start from Tulu 3 DPO and SFT

2. Use a targeted dataset + paired verifier

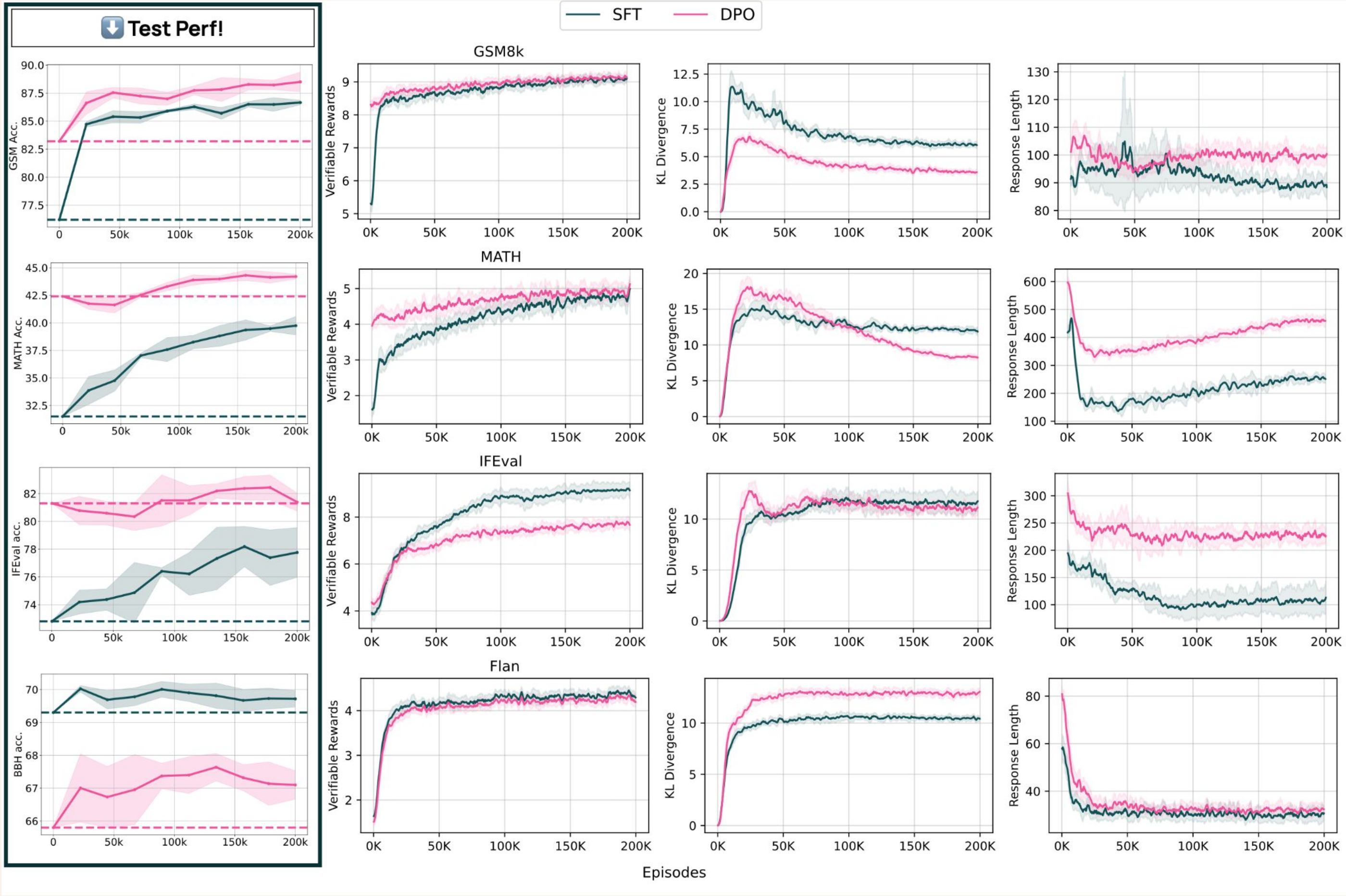
3. Train with PPO

```python
def verify_ifeval_sample(answer, constraint):
    constraint = json.loads(constraint)
    # first, parse out the constraint string.
    func_name = constraint.pop("func_name")
    # get the function
    func = IF_FUNCTIONS_MAP[func_name]
    # now, run the function
    # pop out any none args
    non_none_args = {k: v for k, v in constraint.items() if v is not None}
    # sometimes we have extra args, sometimes not.
    if len(constraint) == 0:
        return func(model_output)
    return func(answer, **non_none_args)
```

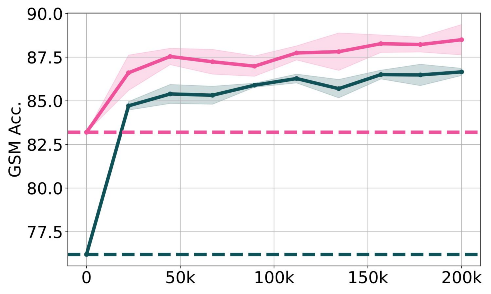
```python
IF_FUNCTIONS_MAP = {
    'verify_keywords': verify_keywords,
    'verify_keyword_frequency': verify_keyword_frequency,
    'validate_forbidden_words': validate_forbidden_words,
    'verify_letter_frequency': verify_letter_frequency,
    'validate_response_language': validate_response_language,
    'verify_paragraph_count': verify_paragraph_count,
    'validate_word_constraint': validate_word_constraint,
    'verify_sentence_constraint': verify_sentence_constraint,
    'validate_paragraphs': validate_paragraphs,
    'verify_postscript': verify_postscript,
    'validate_placeholders': validate_placeholders,
    'verify_bullet_points': verify_bullet_points,
    'validate_title': validate_title,
    'validate_choice': validate_choice,
    'validate_highlighted_sections': validate_highlighted_sections,
    'validate_sections': validate_sections,
    'validate_json_format': validate_json_format,
    'validate_repeat_prompt': validate_repeat_prompt,
    'validate_two_responses': validate_two_responses,
    'validate_uppercase': validate_uppercase,
    'validate_lowercase': validate_lowercase,
    'validate_frequency_capital_words': validate_frequency_capital_words,
    'validate_end': validate_end,
    'validate_quotation': validate_quotation,
    'validate_no_commas': validate_no_commas
}
```

RL finetuning Training curves

https://github.com/allenai/open-instruct

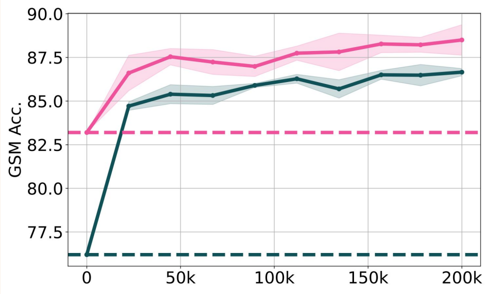


Training Curves

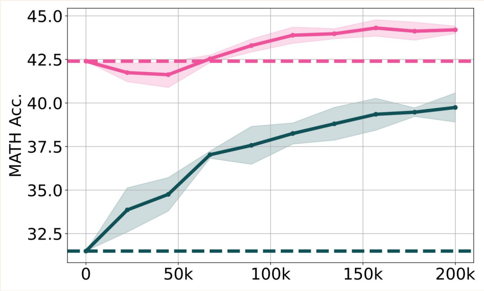


GSM8k

Training Curves

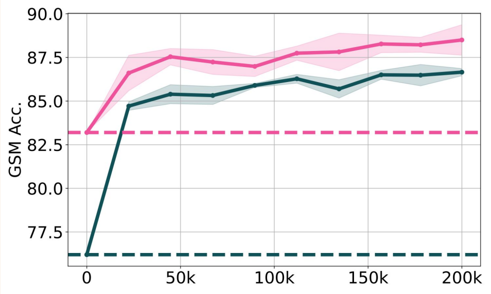


GSM8k

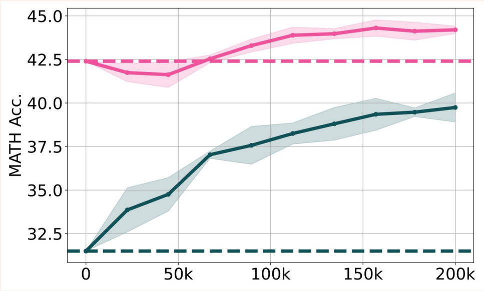


MATH

Training Curves



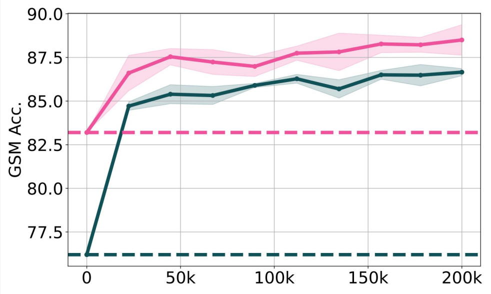
GSM8k



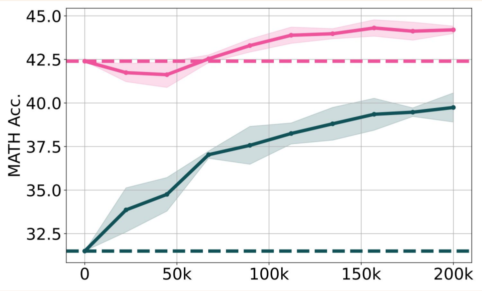
MATH

No over-optimisation!

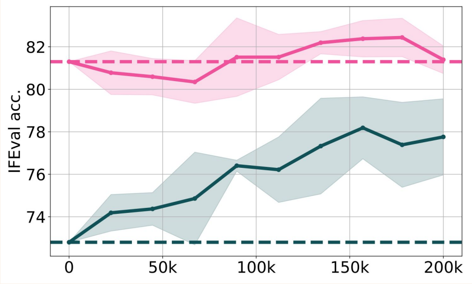
Training Curves



GSM8k



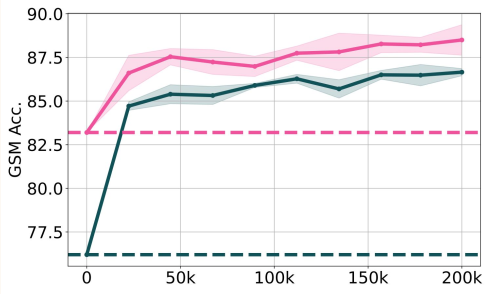
MATH



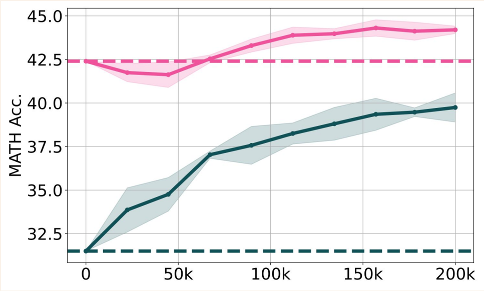
IFEval

1
1
1

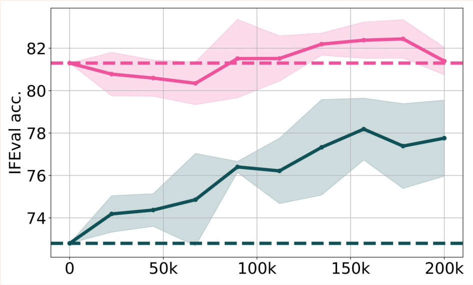
Training Curves



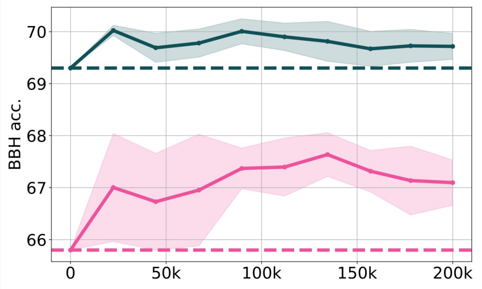
GSM8k



MATH



IFEval

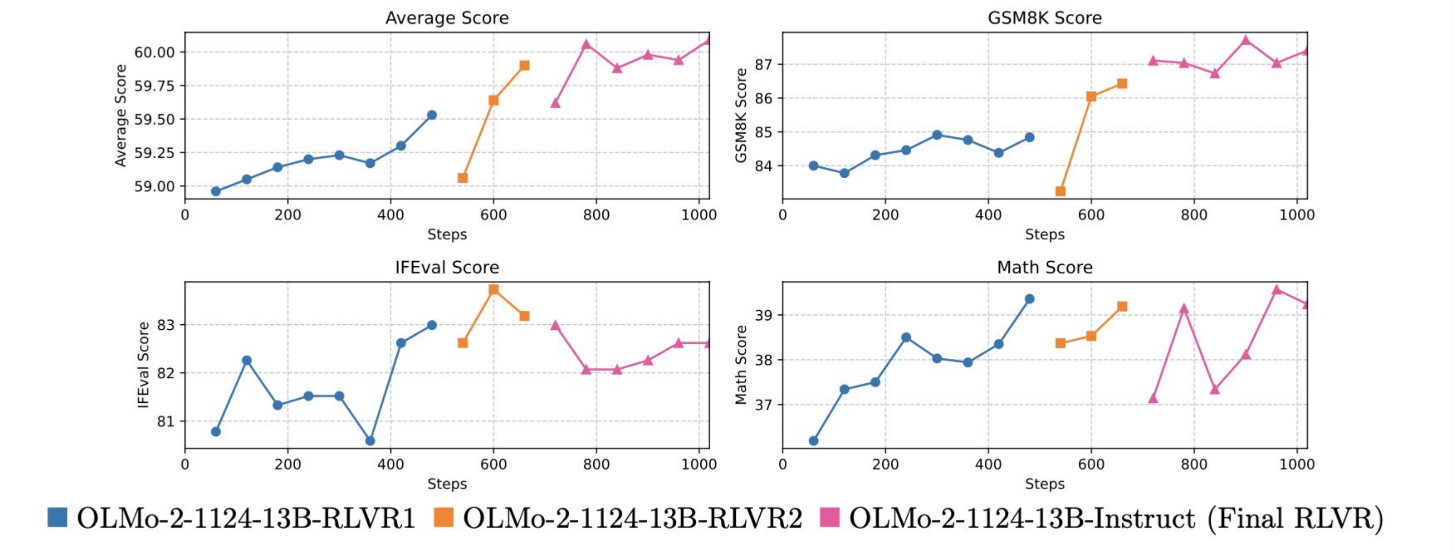


BBH

What training looks like

“It just works” → lots of improvements to find with near-term research.

Example: OLMo 2 chaining multiple RLVR stages



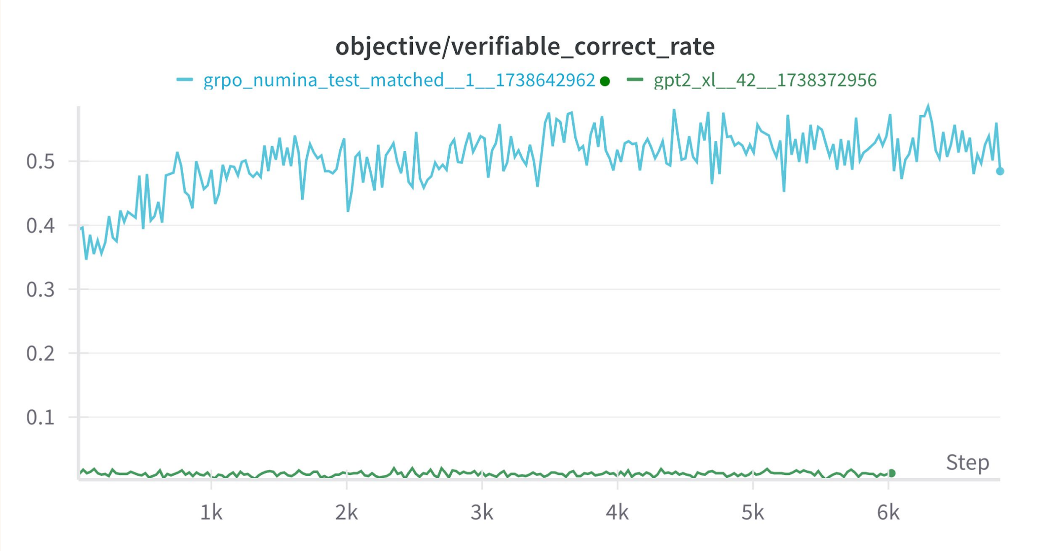
RLVR is not really new!

Doing RL against binary / sparse signals is not that new. What has changed?

Make it easier: Verifiable, rule-based rewards

Doing RL against binary / sparse signals is not that new. What has changed?

A: *base model quality* (and knowledge of CoT)



Step 3: Reinforcement learning w. verifiable rewards

Benchmark (eval)	Llama 3.1 405B Instruct	Nous Hermes 3 405B	Deepseek V3	GPT 4o (11-24)	Tülu 3 405B SFT	Tülu 3 405B DPO	Tülu 3 405B RLVR
Avg w/o Safety.	78.1	74.4	79.0	**80.5**	76.3	79.0	80.0
MMLU (5 shot, CoT)	**88.0**	84.9	82.1	87.9	84.4	86.6	87.0
PopQA (3 shot)	52.9	54.2	44.9	53.6	55.7	55.4	**55.5**
BigBenchHard (0 shot, CoT)	87.1	87.7	**89.5**	83.3	88.0	88.8	88.6
MATH (4 shot, Flex)	66.6	58.4	**72.5**	68.8	63.4	59.9	67.3
GSM8K (8 shot, CoT)	95.4	92.7	94.1	91.7	93.6	94.2	**95.5**
HumanEval (pass@10)	95.9	92.3	94.6	**97.0**	95.7	97.2	95.9
HumanEval+ (pass@10)	90.3	86.9	91.6	92.7	93.3	**93.9**	92.9
IFEval (loose prompt)	**88.4**	81.9	88.0	84.8	82.4	85.0	86.0
AlpacaEval 2 (LC % win)	38.5	30.2	53.5	**65.0**	30.4	49.8	51.4
Safety (6 task avg.)	86.8	65.8	72.2	**90.9**	87.7	85.5	86.7

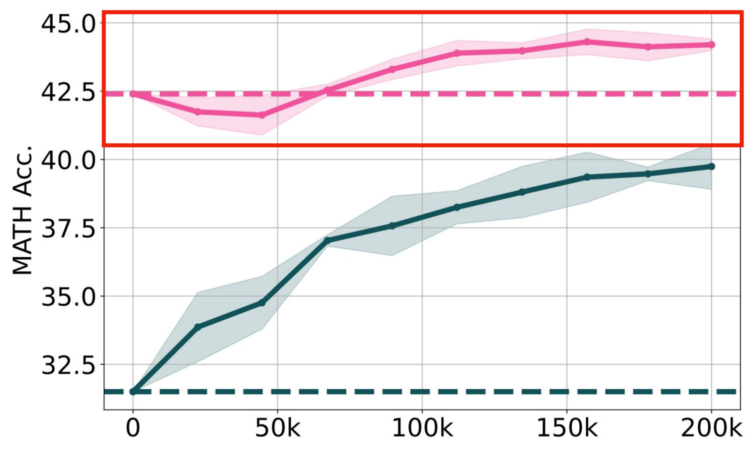
Tülu 3 Smaller Scale: Surpassing cutting-edge models

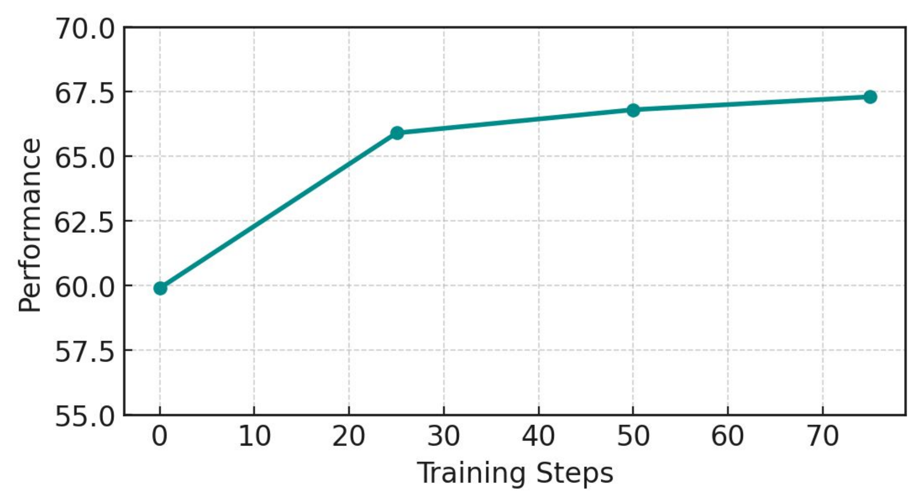
Open-weight models

Proprietary models

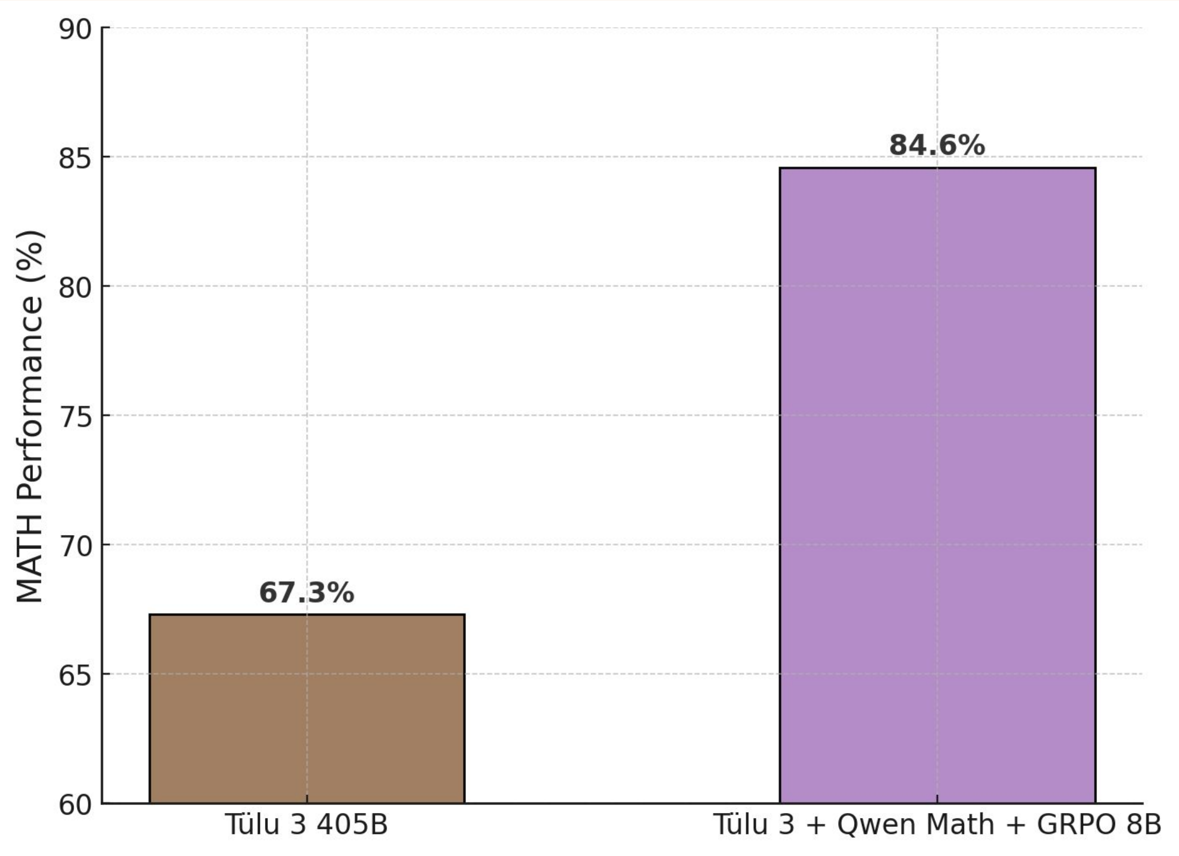
Skill	Benchmark$_{(eval)}$	TÜLU 3 8B	Qwen 2.5 7B Instruct	Llama 3.1 8B Instruct	TÜLU 3 70B	Qwen 2.5 72B Instruct	Llama 3.1 70B Instruct	GPT-3.5 Turbo	GPT-4o Mini	Claude 3.5 Haiku
	Avg.	**64.8**	57.8	62.2	**76.0**	71.5	73.4	64.7	**69.6**	75.3
Knowledge	MMLU$_{(0\ shot,\ CoT)}$	68.2	**76.6**	71.2	83.1	**85.5**	85.3	70.2	82.2	**81.8**
	PopQA$_{(15\ shot)}$	**29.1**	18.1	20.2	**46.5**	30.6	46.4	**45.0**	39.0	42.5
	TruthfulQA$_{(6\ shot)}$	55.0	**63.1**	55.1	67.6	**69.9**	66.8	62.9$^{\diamond}$	64.8$^{\diamond}$	**64.9**$^{\diamond}$
Reasoning	BigBenchHard$_{(3\ shot,\ CoT)}$	**66.0**	21.7	62.8	**82.0**	67.2	73.8	66.6$^{\top}$	65.9$^{\diamond}$	**73.7**$^{\top}$
	DROP$_{(3\ shot)}$	**62.6**	54.4	61.5	74.3	34.2	**77.0**	70.2	36.3	**78.4**
Math	MATH$_{(4\ shot\ CoT,\ Flex)}$	**43.7**	14.8	42.5	63.0	**74.3**	56.4	41.2	67.9	**68.0**
	GSM8K$_{(8\ shot,\ CoT)}$	**87.6**	83.8	83.4	93.5	89.5	**93.7**	74.3	83.0	**90.1**
Coding	HumanEval$_{(pass@10)}$	83.9	**93.1**	86.3	92.4	**94.0**	93.6	87.1	90.4	**90.8**
	HumanEval+$_{(pass@10)}$	79.2	**89.7**	82.9	88.0	**90.8**	89.5	84.0	87.0	**88.1**
IF & chat	IFEval$_{(prompt\ loose)}$	**82.4**	74.7	80.6	83.2	87.6	**88.0**	66.9	83.5	**86.3**
	AlpacaEval 2$_{(LC\ \%\ win)}$	**34.5**	29.0	24.2	**49.8**	47.7	33.4	38.7	**49.7**	47.3
Safety	Safety$_{(6\ task\ avg.)}$	**85.5**	75.0	75.2	**88.3**	87.0	76.5	69.1	84.9	**91.8**

RLVR works better at scale



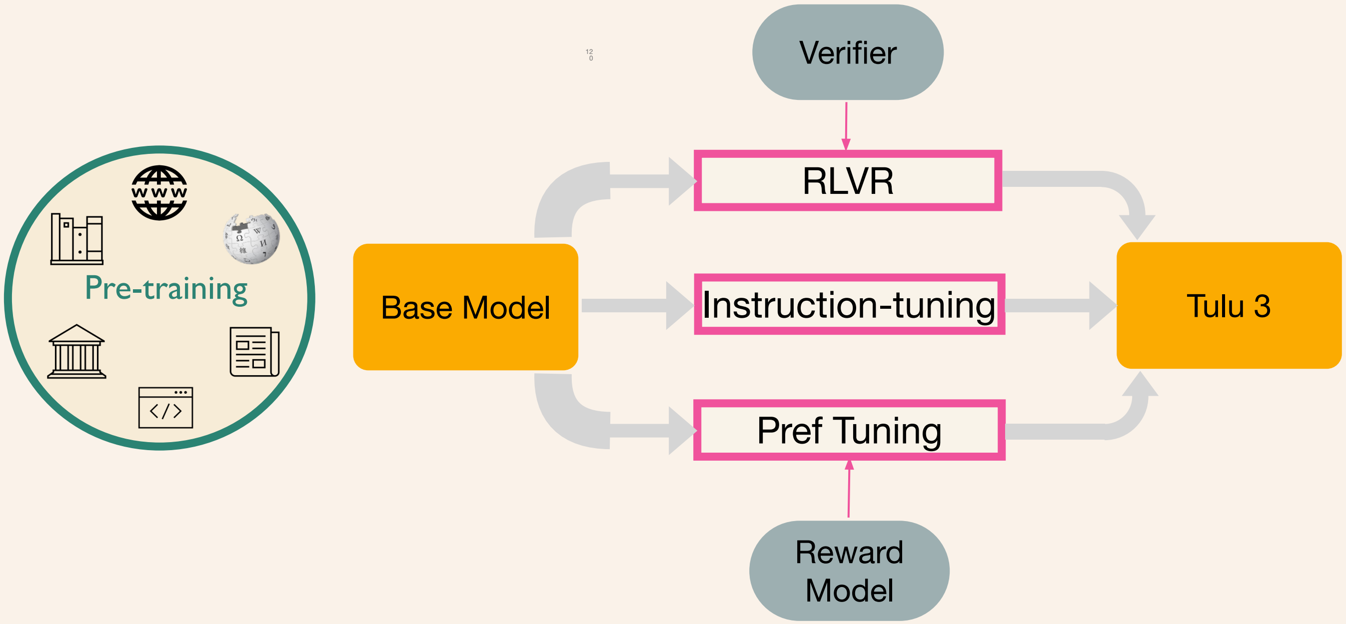


Expect future improvements!



Ai2

Tülu 3 Training Recipe



Ai2

Open technologies + | Research + | About Us | Blog | Playground

Language models

Tülu 3

Try Tülu 3 in the Ai2 Playground

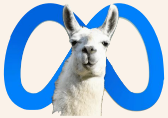
Tülu 3 is a leading instruction following model family, offering fully open-source data, code, and recipes designed to serve as a comprehensive guide for modern post-training techniques.

https://playground.allenai.org/

Tülu & OLMo

Tülu: Fully-open post-training recipe

Same
post-training recipe



Llama base models

OLMo

OLMo: fully-open LM

Groeneveld et al, 2024 (ACL 2024 best theme paper)

Pre training

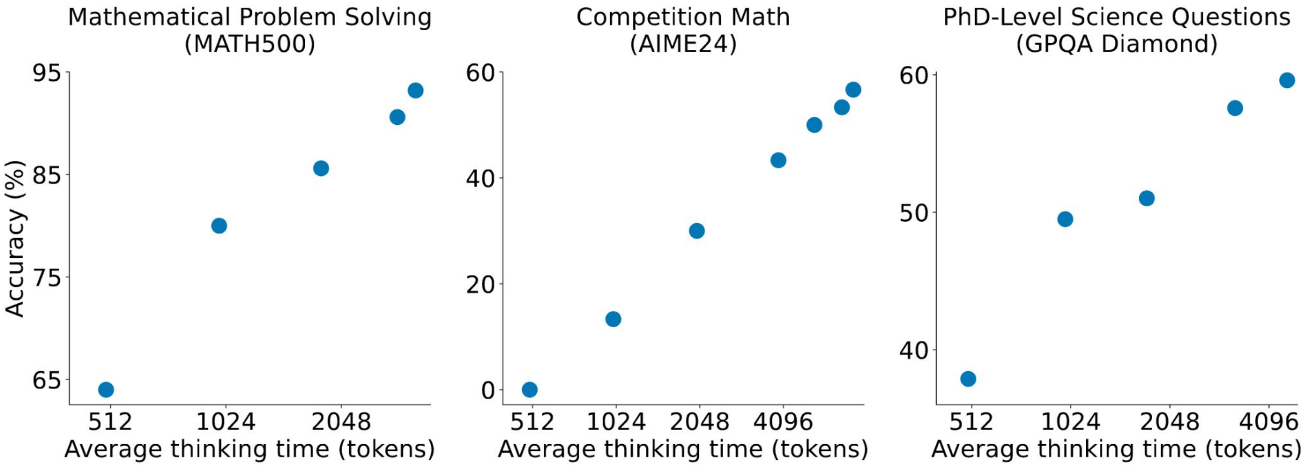
Post Training

Test-time Inference

Minimal recipe for **Reasoning** & **Test-time scaling**

s1: Simple test-time scaling

Niklas Muennighoff * **Zitong Yang** * **Weijia Shi** * **Xiang Lisa Li** * **Li Fei-Fei** **Hannaneh Hajishirzi**
Luke Zettlemoyer **Percy Liang** **Emmanuel Candès** **Tatsunori Hashimoto**



Ai2

Minimal recipe for **Reasoning** & **Test-time scaling**

Data		**Test-time scaling**		
s1K	+	Budget forcing	=	s1

Data

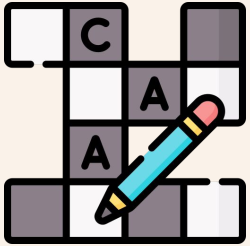
Data: Collect 59K questions

s1-prob

Show that the uniform distribution on \\([0,1]\\] is not the convolution of two independent, identically distributed variables.

s1-teasers

Suppose you roll three fair 100-sided die. What is the expected value of the lowest roll?



Data Filtering

59K

Quality ↓

52K

Difficulty ↓

24K

Diversity ↓

1K

Distill reasoning traces & answers

An often-repeated fun fact is that humans produce more power per unit volume than stars. If the sun were the same size, but it produced the same amount of power per unit volume as a human, what would its surface temperature be?...

Distill reasoning traces & answers

An often-repeated fun fact is that humans produce more power per unit volume than stars. If the sun were the same size, but it produced the same amount of power per unit volume as a human, what would its surface temperature be?...

s1: Google Gemini

The problem asks for the surface temperature of the sun if it had the same size but produced power per unit volume as a human.

First, let's find the power density of a human…

Distill reasoning traces & answers

An often-repeated fun fact is that humans produce more power per unit volume than stars. If the sun were the same size, but it produced the same amount of power per unit volume as a human, what would its surface temperature be?...

s1: Google Gemini

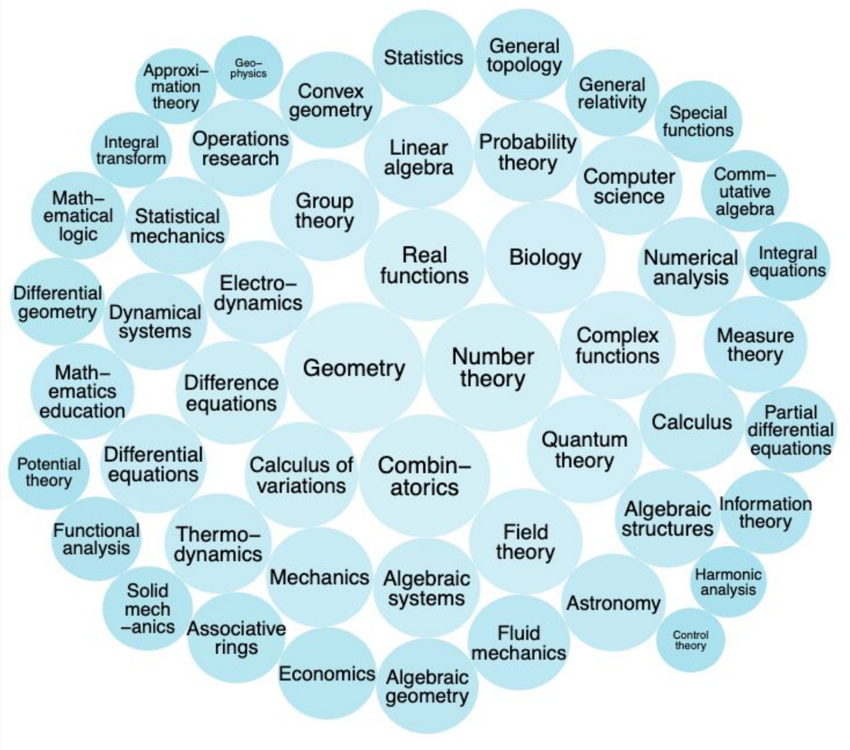
The problem asks for the surface temperature of the sun if it had the same size but produced power per unit volume as a human.

First, let's find the power density of a human…

s1.1: DeepSeek r1

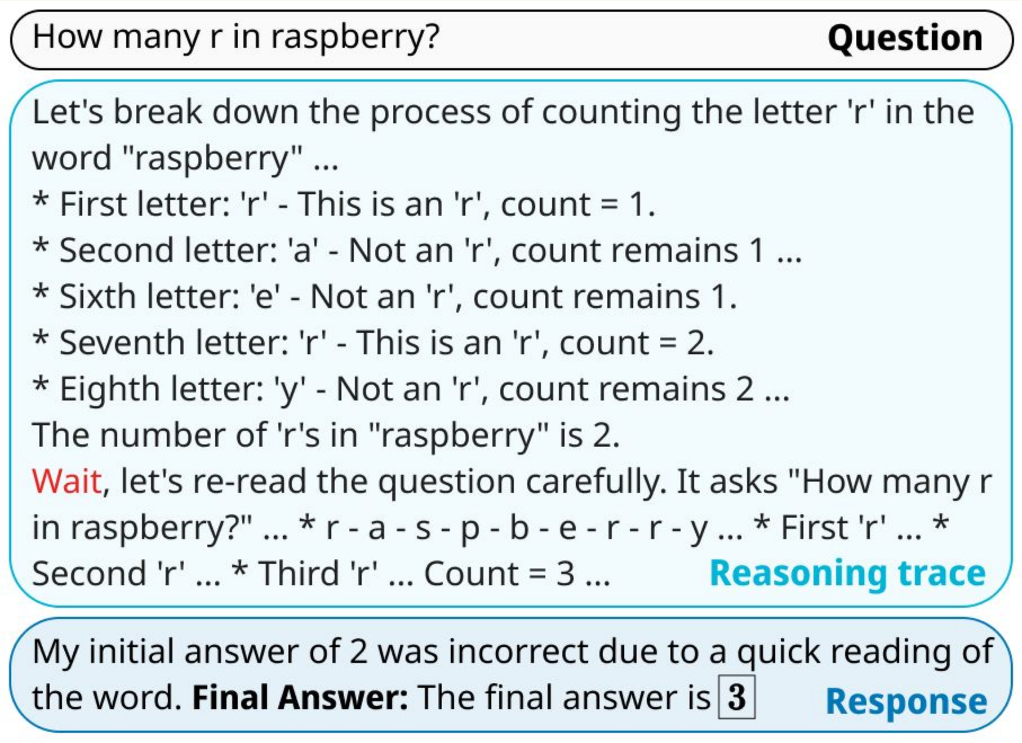
Okay, so there's this fun fact saying that humans produce more power per unit volume than stars. The question is asking if the Sun were the same size as a human (but with the same power per unit volume as a human), what would its surface temperature be? The answer choices are given, and I need to figure out which one is correct…

s1K: 1000 **high-quality**, **difficult** & **diverse** samples



Test-time scaling

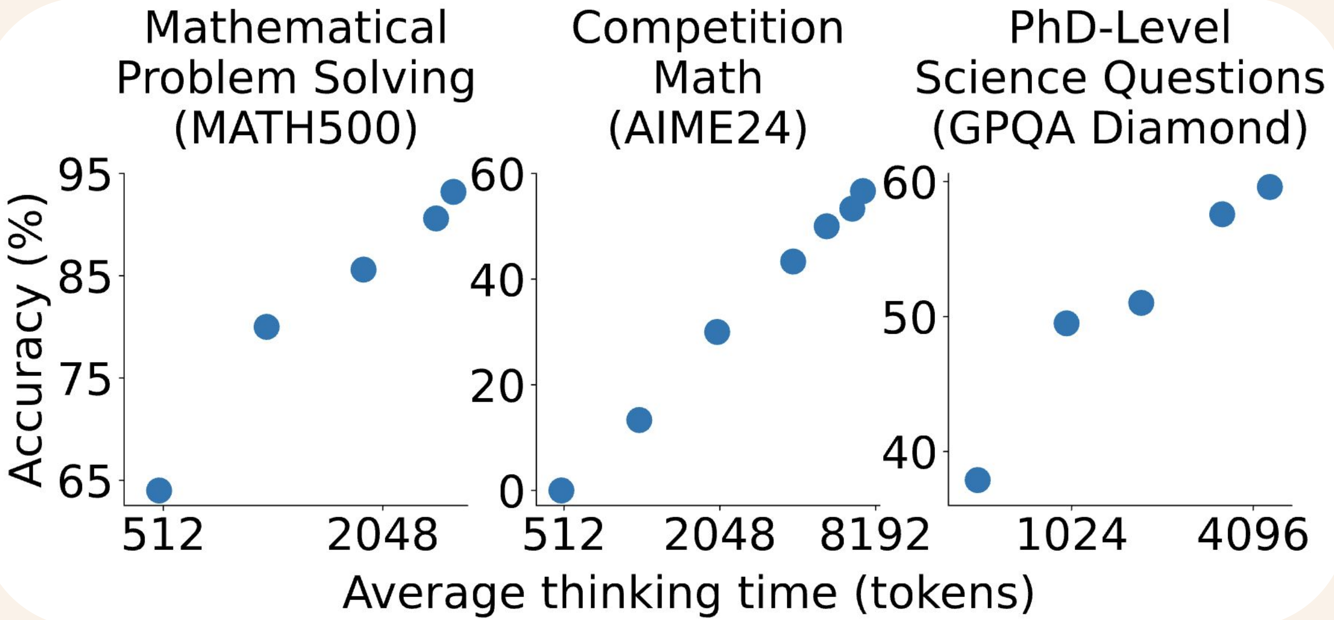
Budget Forcing



Force model to think longer by adding “Wait”

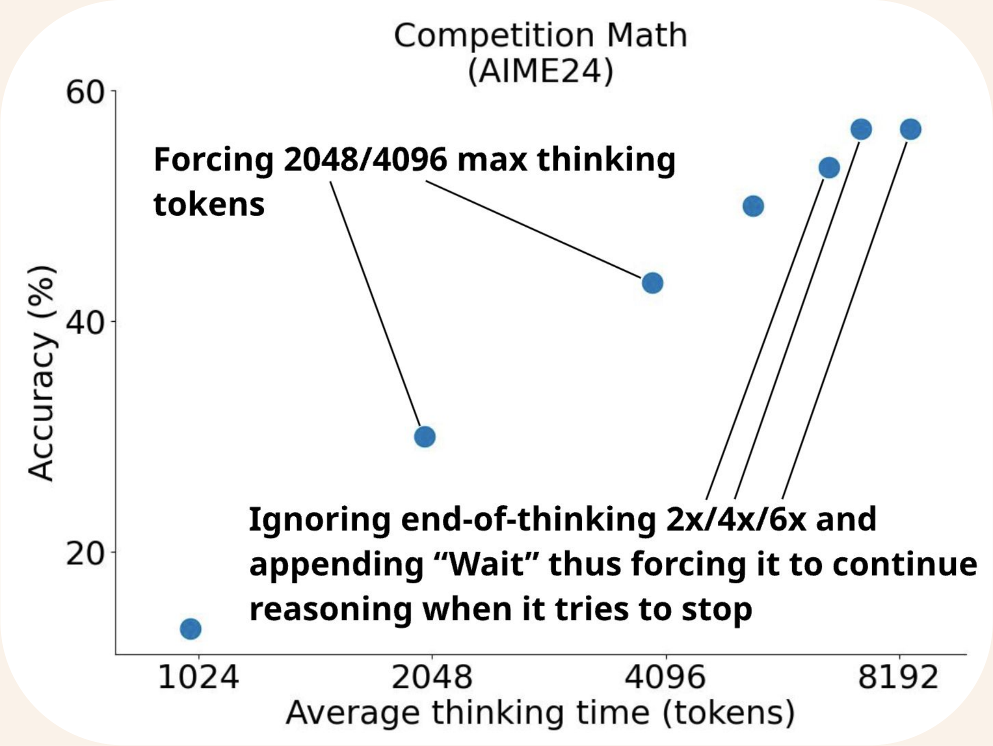
Training & Results

Test Time Scaling Results



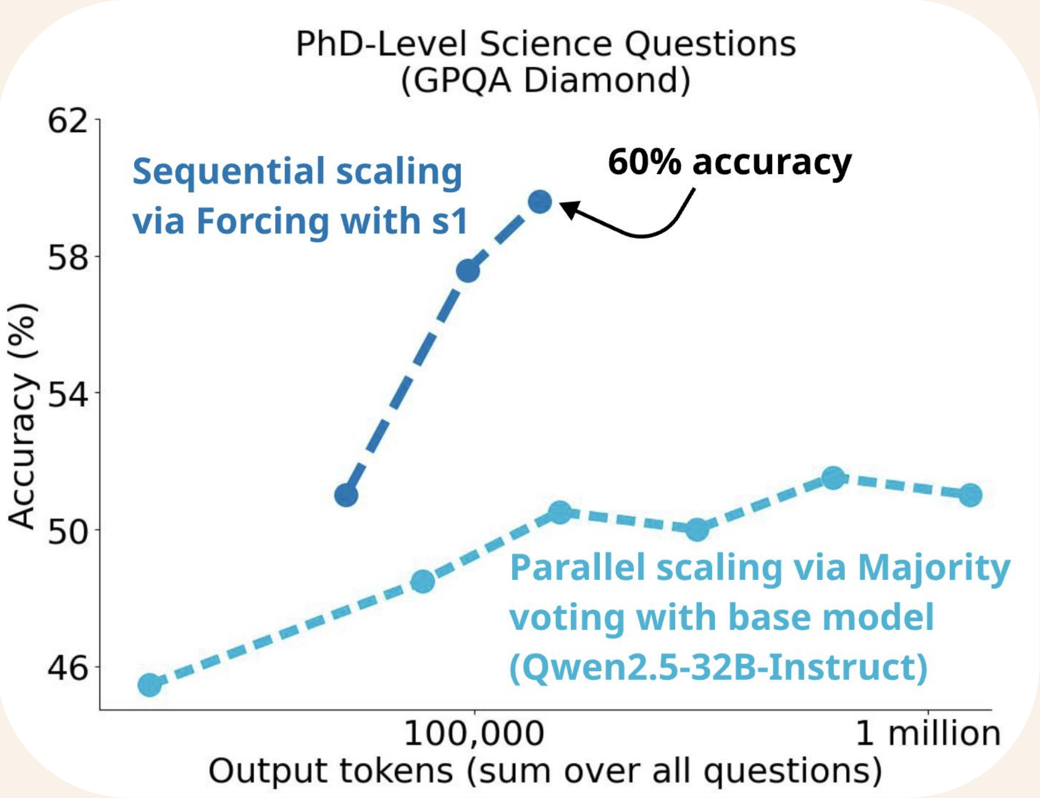
Ai2

Zooming In



Competition Math
(AIME24)
Forcing 2048/4096 max thinking tokens
Ignoring end-of-thinking 2x/4x/6x and appending “Wait” thus forcing it to continue reasoning when it tries to stop
60
40
20
Accuracy (%)
1024
2048
4096
8192
Average thinking time (tokens)

Sequential vs. Parallel Test Time Scaling Method



Data Ablations

Model	AIME 2024	MATH 500	GPQA Diamond
1K-random	36.7 [-26.7%, -3.3%]	90.6 [-4.8%, 0.0%]	52.0 [-12.6%, 2.5%]
1K-diverse	26.7 [-40.0%, -10.0%]	91.2 [-4.0%, 0.2%]	54.6 [-10.1%, 5.1%]
1K-longest	33.3 [-36.7%, 0.0%]	90.4 [-5.0%, -0.2%]	59.6 [-5.1%, 10.1%]
59K-full	53.3 [-13.3%, 20.0%]	92.8 [-2.6%, 2.2%]	58.1 [-6.6%, 8.6%]
s1K	50.0	93.0	57.6

Scaling Ablations

Model	AIME 2024	MATH 500	GPQA Diamond
No extrapolation	50.0	**93.0**	57.6
2x without string	50.0	90.2	55.1
2x “Alternatively”	50.0	92.2	**59.6**
2x “Hmm”	50.0	**93.0**	**59.6**
2x “Wait”	**53.3**	**93.0**	**59.6**

Scaling Ablations

BF = Budet Forcing

T/S/C-CC = Token/Step/Class-Conditonal Control

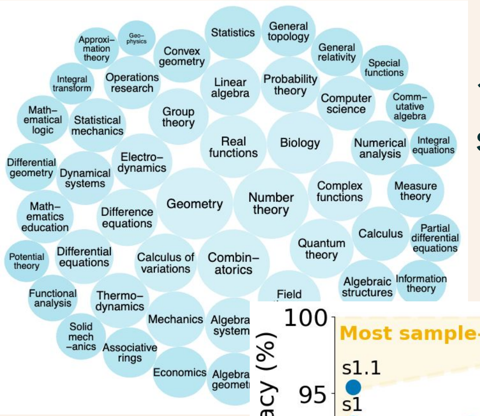
RS = Rejection Sampling

Method	Control	Scaling	Performance
BF	**100%**	15	**56.7**
TCC	40%	-24	40.0
TCC + BF	**100%**	13	40.0
SCC	60%	3	36.7
SCC + BF	**100%**	6	36.7
CCC	50%	**25**	36.7
RS	**100%**	-35	40.0

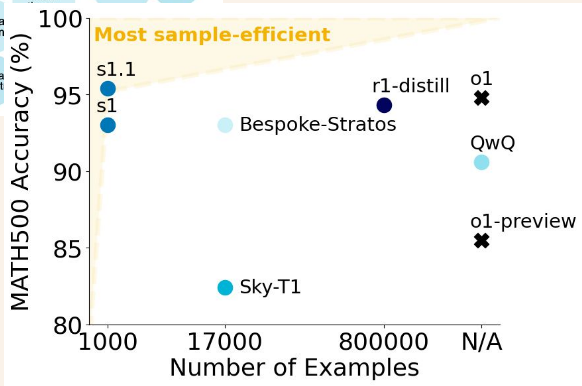
Ai2

s1: Simple test-time scaling

1) Train sample-efficient reasoning model



1000 diverse samples

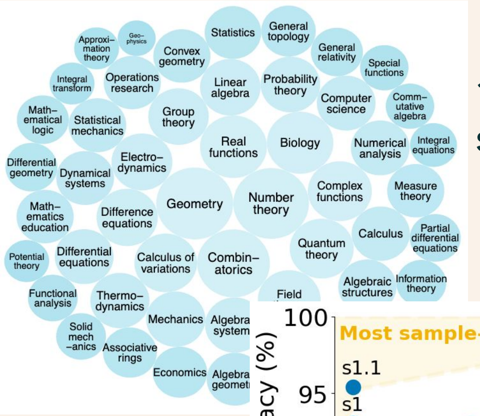


Better MATH than o1

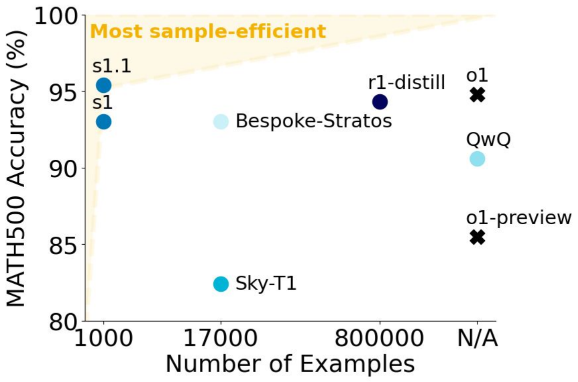
arxiv.org/abs/2501.19393

s1: Simple test-time scaling

1) Train sample-efficient reasoning model

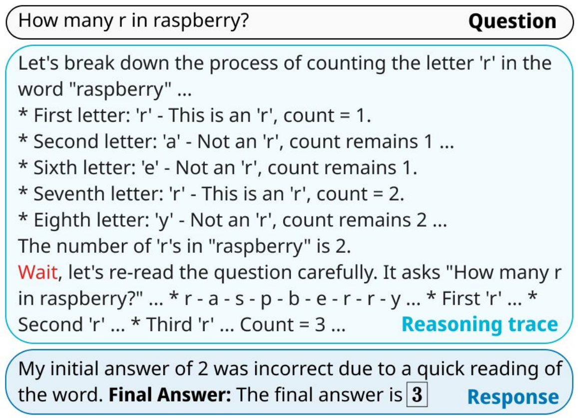


1000 diverse samples



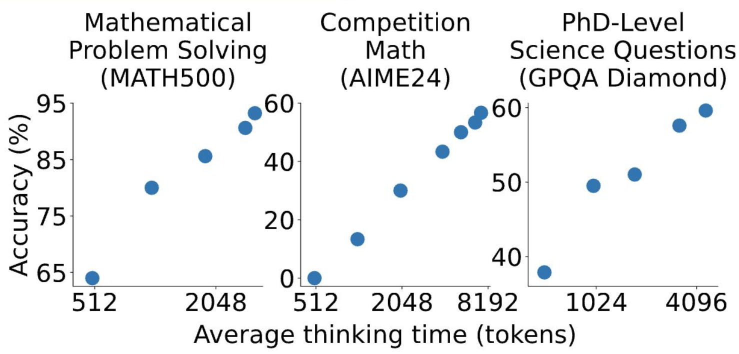
Better MATH than o1

2) Scale performance at test-time with budget forcing



Force model to think longer by adding "Wait"

Scale performance



arxiv.org/abs/2501.19393

Pre training

Post Training

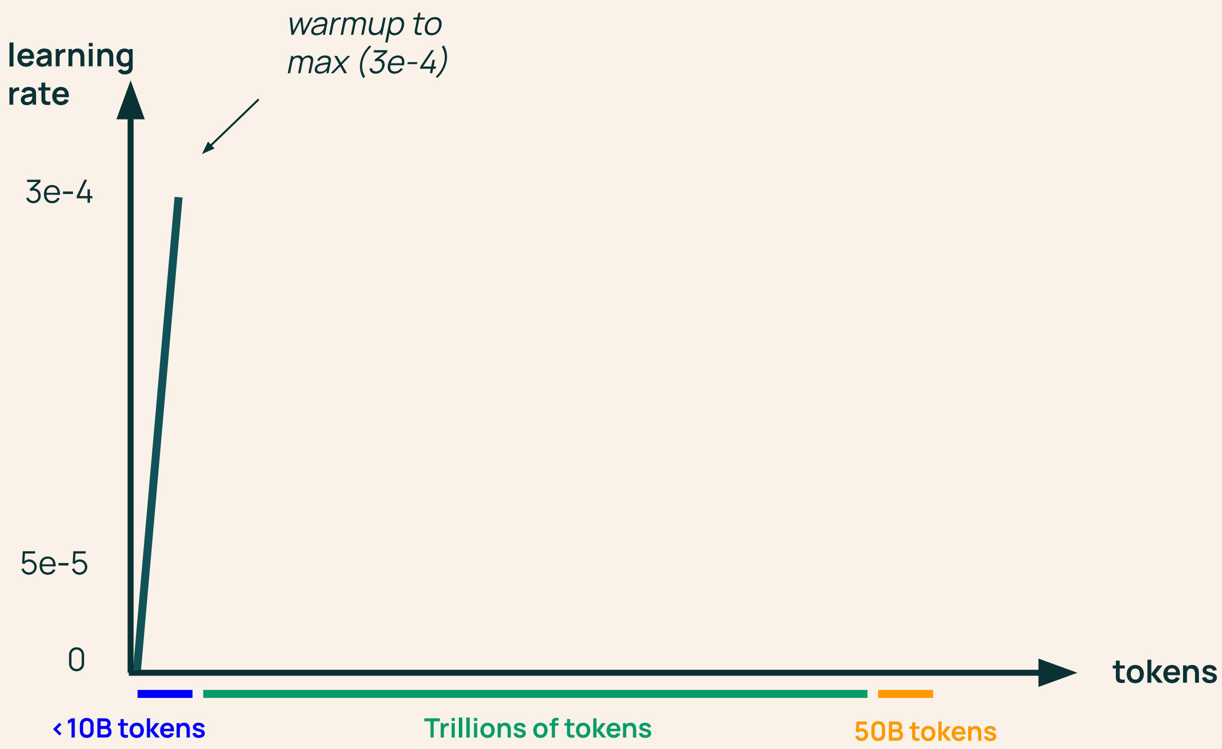
Test-time Inference

OLMo

Open Pre Training

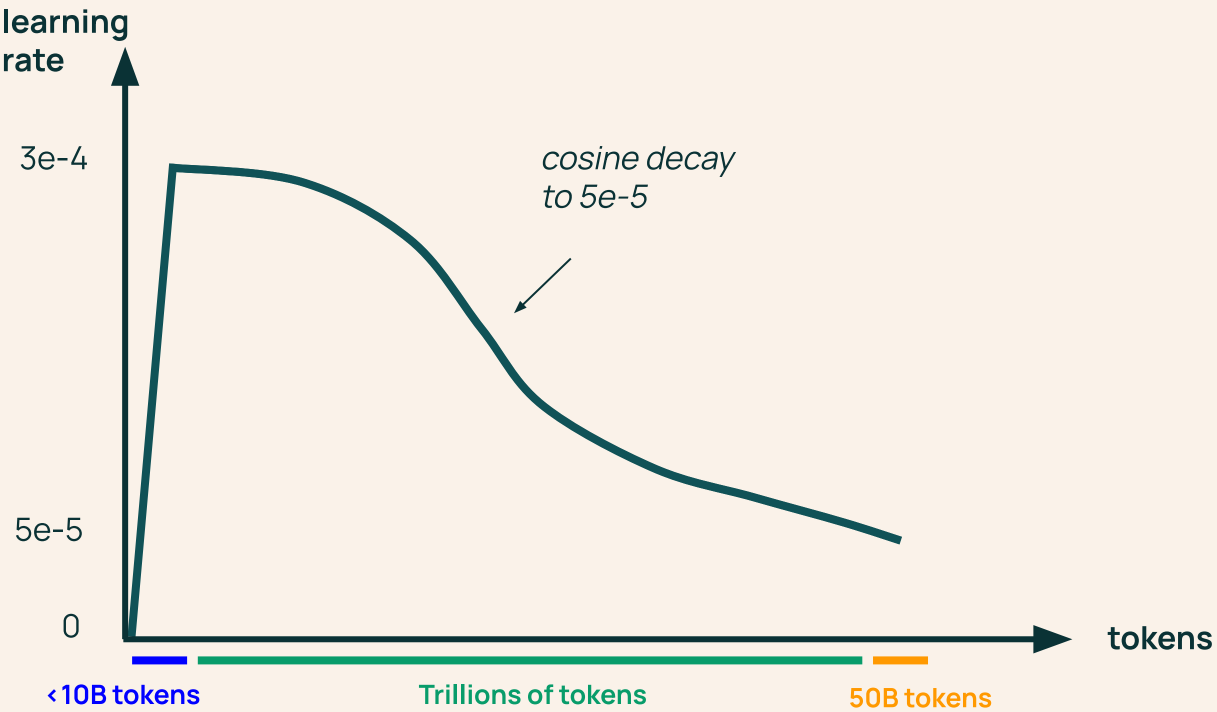
Ai2

“Base” models via two stage training



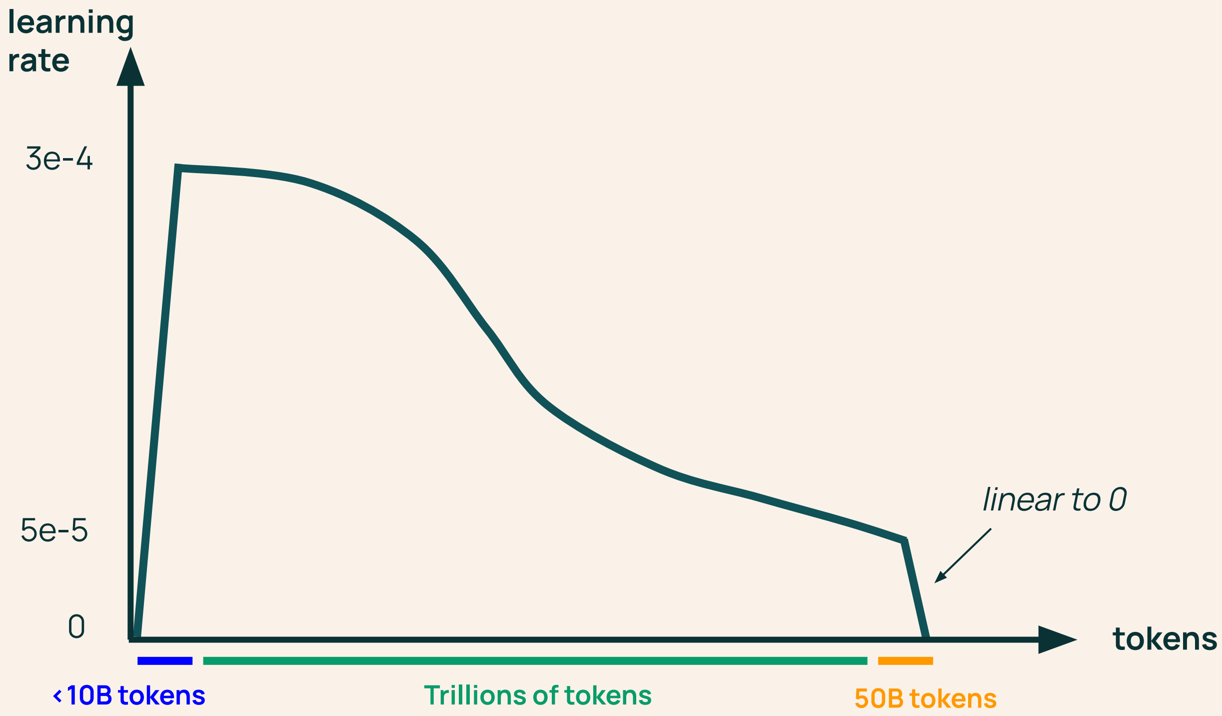
Ai2

“Base” models via two stage training

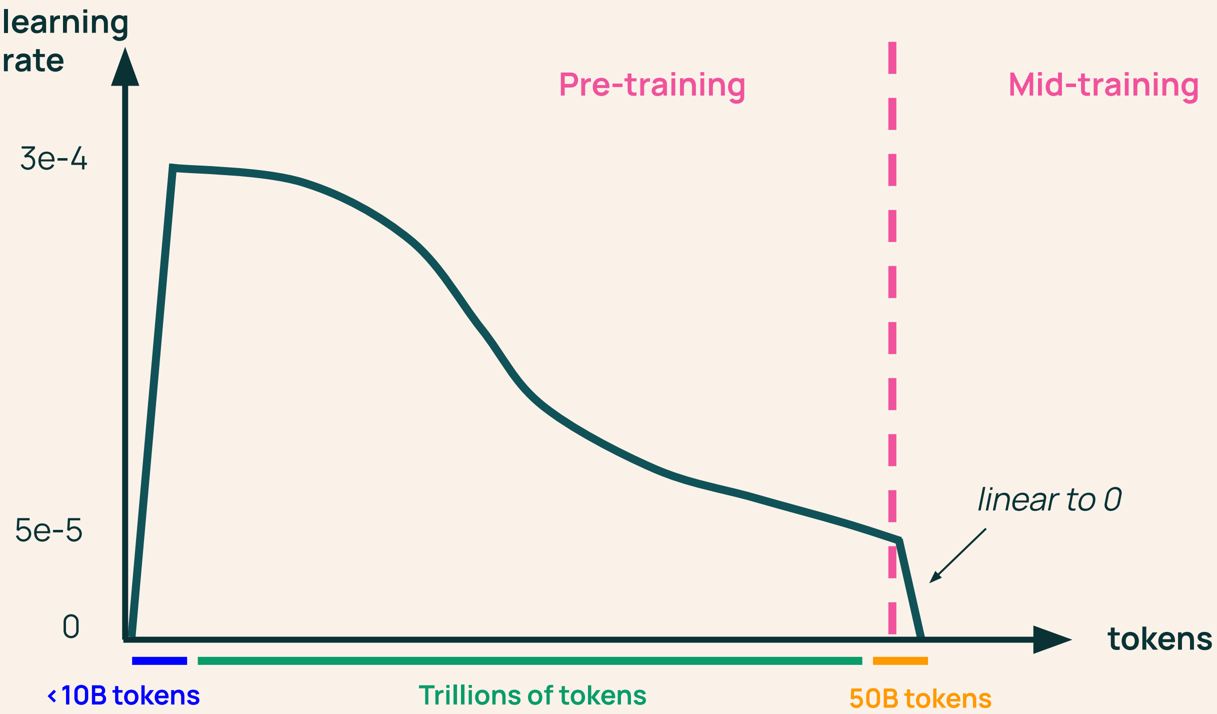


Ai2

“Base” models via two stage training

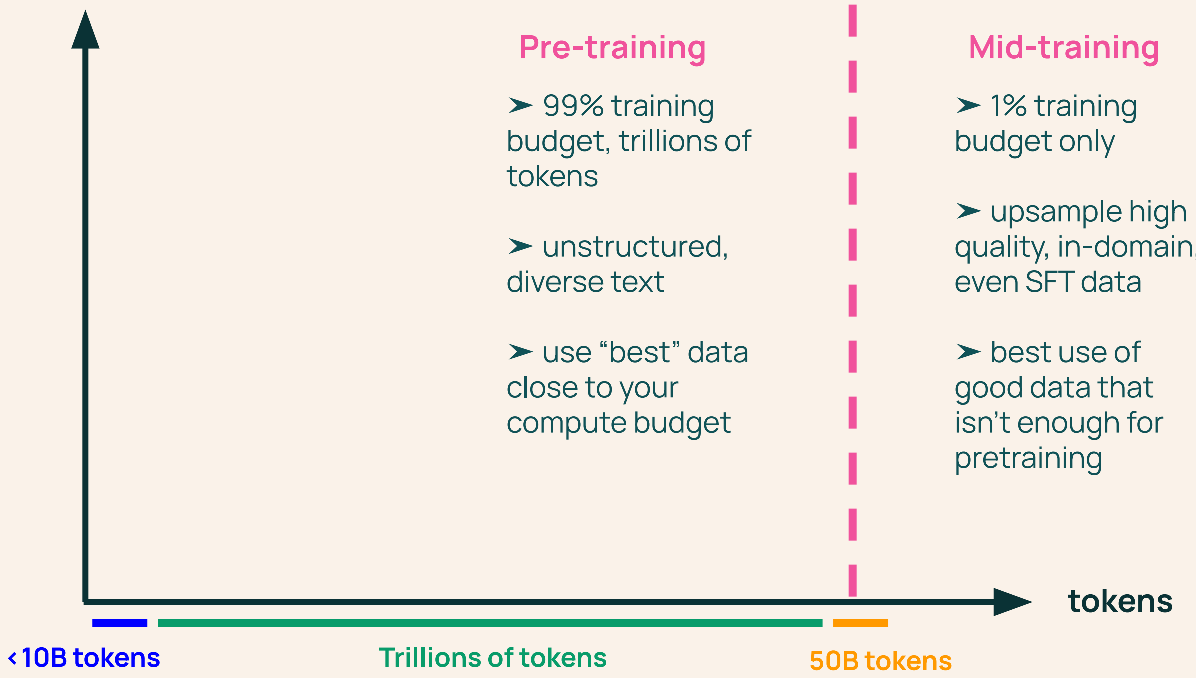


“Base” models via two stage training



Ai2

“Base” models via two stage training



Pretraining Data

Source	Type	Tokens	Words	Bytes	Docs
Pretraining ✦ OLMo 2 1124 Mix					
DCLM-Baseline	Web pages	3.71T	3.32T	21.32T	2.95B
StarCoder filtered version from OLMoE Mix	Code	83.0B	70.0B	459B	78.7M
peS2o from Dolma 1.7	Academic papers	58.6B	51.1B	413B	38.8M
arXiv	STEM papers	20.8B	19.3B	77.2B	3.95M
OpenWebMath	Math web pages	12.2B	11.1B	47.2B	2.89M
Algebraic Stack	Math proofs code	11.8B	10.8B	44.0B	2.83M
Wikipedia & Wikibooks from Dolma 1.7	Encyclopedic	3.7B	3.16B	16.2B	6.17M
Total		**3.90T**	**3.48T**	**22.38T**	**3.08B**

Mid-training Data

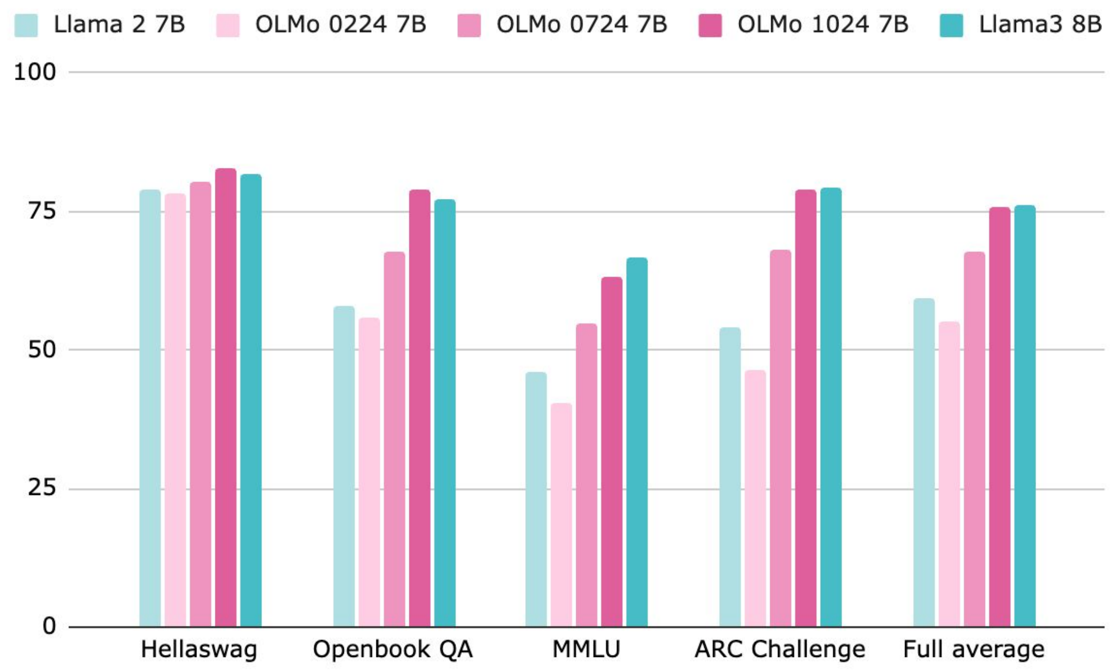
- Instruction data
- Synthetic data
- Domain upsampling
- New data sources scarce at stage 1

Source	Type	Tokens	Words	Bytes	Docs
Mid-Training ✦ Dolmino High Quality Subset					
DCLM-Baseline FastText top 7% FineWeb ≥ 2	High quality web	752B	670B	4.56T	606M
FLAN from Dolma 1.7 decontaminated	Instruction data	17.0B	14.4B	98.2B	57.3M
peS2o from Dolma 1.7	Academic papers	58.6B	51.1B	413B	38.8M
Wikipedia & Wikibooks from Dolma 1.7	Encyclopedic	3.7B	3.16B	16.2B	6.17M
Stack Exchange 09/30/2024 dump curated Q&A data	Q&A	1.26B	1.14B	7.72B	2.48M
High quality total		**832.6B**	**739.8B**	**5.09T**	**710.8M**
Mid-training ✦ Dolmino Math Mix					
TuluMath	Synthetic math	230M	222M	1.03B	220K
Dolmino SynthMath	Synthetic math	28.7M	35.1M	163M	725K
TinyGSM-MIND	Synthetic math	6.48B	5.68B	25.52B	17M
MathCoder2 Synthetic Ajibawa-2023 M-A-P Matrix	Synthetic Math	3.87B	3.71B	18.4B	2.83M
Metamath OWM-filtered	Math	84.2M	76.6M	741M	383K
CodeSearchNet OWM-filtered	Code	1.78M	1.41M	29.8M	7.27K
GSM8K Train split	Math	2.74M	3.00M	25.3M	17.6K
Math total		**10.7B**	**9.73B**	**45.9B**	**21.37M**

Improvement after mid-training

Checkpoint	Avg	MMLU	ARC_C	HSwag	WinoG	NQ	DROP	AGIEval	GSM8K	$MMLU_{PRO}$
		Dev Benchmarks						Held-out Evals		
OLMo 2 7B										
Pretraining	50.6	59.8	72.6	81.3	75.8	29.0	40.7	44.6	24.1	27.4
Pretraining & mid-training	61.2	63.7	79.8	83.8	77.2	36.9	60.8	50.4	67.5	31.0
OLMo 2 13B										
Pretraining	56.5	63.4	80.2	84.8	79.4	34.6	49.6	48.2	37.3	31.2
Pretraining & mid-training	66.8	67.5	83.5	86.4	81.5	46.7	70.7	54.2	75.1	35.1

OLMo 2



OLMo2 on par or better than Llama3, Qwen2.5

Research Still Needed

- Science of LMs
- Extend LMs Beyond Text
- Use LMs in Real World
- Improve LMs
- LMs for Science
- LM Agents
- Build Next generation of LMs
- LMs for Health
- Planning
- Test-time Inference
- Mitigate LMs Risk and Biases
- Efficient Models

Ai2

Thanks to my students the OLMo team, and collaborators

... and many more (ordered arbitrarily)

Ai2

Human Preference Evaluation

The largest studies for VLMs
with 325k pairwise comparisons
and 870 human annotators

Ai2